AIAA/AHS
Flight Simulation Technologies Conference

A
COLLECTION
OF
TECHNICAL
PAPERS

Hilton Head Island, SC
August 24-26, 1992

AIAA

For permission to copy or republish, contact:
American Institute of Aeronautics and Astronautics
The Aerospace Center
370 L'Enfant Promenade, SW
Washington, DC 20024-2518

A COLLECTION OF TECHNICAL PAPERS

AIAA/AHS
Flight Simulation Technologies Conference

August 24-26, 1992/Hilton Head Island, SC

Printed in the U.S.A.

INTRODUCTORY REMARKS

It is my pleasure to introduce the Collection of Technical Papers for the 1992 AIAA Flight Simulation Technology Conference.

These papers represent the state of Simulation technology during a period of great change in the Aerospace industry. Included in this collection are papers which describe innovations in technology to lower cost, increase flexibility, reduce schedules and improve the quality in the programs of which they are a part. These innovations, and the uses to which simulators are put, continue to demonstrate the value of the simulator in the aerospace design process.

This conference was co-sponsored by the American Helicopter Society, which accounts for the large amount of papers which address rotorcraft applications. This co-sponsorship has been both enjoyable and technically fruitful.

There is considerable synthesis between the papers in this collection, and those of the associated collections for the AIAA Aircraft Design Systems Meeting and the AIAA Biennial Flight Conference. Papers in this collection reference aircraft design and flight test programs, and papers in the other collections make many references to flight simulation. A review of all three collections provides a thorough overview of these three disciplines as they are to be found in the early nineties.

Particular appreciation is due Ms. Ann Wildgen, AIAA Technical Chairperson, for her excellent efforts in organizing the technical program. I would also like to express my appreciation to Mr. John Johns, who acted as the AHS Technical Chairperson, and whom was responsible for liaison with AHS and for attracting the excellent slate of rotorcraft simulation papers. Appreciation is also extended to the session chairpersons, and of course to Ms. Eleni Kleamis, AIAA, for her efficient and professional management of the preparation of the conference.

Matt Landry
1992 Flight Simulation Technology Conference
General Chairman

AIAA/AHS Flight Simulation Technologies Conference
Hilton Head Island, SC/August 24-26, 1992

Table of Contents

N/A - Not Available

W/D - Withdrawn

Simulator Architecture (6)

Rotorcraft Modeling (7)

Simulation Networking I (8)

N/A - Not Available

W/D - Withdrawn

Motion Systems (9)

Operations and Management (11)

Pilot Cuing (12)

N/A - Not Available W/D - Withdrawn

Simulation Networking II (13)

In-Flight Simulation (14)

Simulation Facilities (15)

N/A - Not Available

W/D - Withdrawn

AIAA-92-4130-CP

SHIPBOARD MISSION TRAINING EFFECTIVENESS OF THE NAVAL AIR WARFARE CENTER'S V-22 GOVERNMENT TEST PILOT TRAINER

C. Miller
G. Vandervliet
Rotary Wing Aircraft Test Directorate / Strike Aircraft Test Directorate
Naval Air Warfare Center, Aircraft Division
Patuxent River, Maryland 20670
U.S.A.

Abstract

Initial shipboard compatibility tests of the V-22 Osprey VSTOL tilt-rotor aircraft were conducted aboard the USS Wasp (LHD-1) on 4-8 December 1990. Pilot training and engineering analysis for the first shipboard launches and recoveries of the V-22 Osprey aircraft were conducted in the V-22 simulator at the Naval Air Test Center. Workload data was recorded and the pilots made comments regarding the quality of the training during these sessions. Quantitative comparisons of time history data were made using power spectral densities, averages and maximum attained values. Qualitative comments recorded during training were compared to the remarks made after the actual at-sea tests. The data help conclude that the training was effective, and point to possible improvements in the system for future sessions.

List of Symbols

IGE	In ground effect
OGE	Out of ground effect
PSD	Power spectral density
Hz	Frequency, Hertz

1.0 Background

1.1 Helicopter launch and recovery aboard ship are acts which impose great demands on both the aviator and the aircraft. They are tasks in which success depends on factors that are inherent to both the aircraft and the ship. The helicopter is particularly unfortunate, as it is a fundamentally unstable machine that is required to operate under the most adverse of shipboard conditions to complete its missions. The hazards associated with shipboard operations have forced the helicopter test and evaluation community to regard these operations with great respect.

1.2 In the past, the U.S. Navy's testing of helicopter shipboard operations came long after most of the Navy's helicopters were acquired. Through years of trial and error, helicopter aircrews have learned shipboard lessons the hard way. However, the U.S. Navy's V-22 Osprey Full Scale Development (FSD) program required its aircraft prototypes to have the capability to test for shipboard compatibility prior to production. This was the first FSD aircraft program to undertake a shipboard test so early in the development cycle. The shipboard testing carried out was only the second government test series of the aircraft.

1.3 Conducting shipboard tests early in an aircraft development cycle heightens the hazards of the test, as the aircraft itself is still undergoing a developmental process. This is where the powerful tool of simulation becomes irreplaceable if and only if the simulator provides the required level of fidelity with regards to both the aircraft and the environment models.

1.4 The United States Navy has constructed and fielded the V-22 Government Test Pilot Trainer which is a high fidelity training and flight test support simulation. This simulator is the primary training device for the developmental and operational V-22 test pilots. This training system has been continually upgraded to support the training of V-22 crews in the required procedures for shipboard launch and recovery operations as the FSD program progresses.

1.5 The effectiveness and quality of the training received from simulator flights can be partially evaluated by comparing quantitative and qualitative data of similar parameters from similar tasks flown during simulator training flights and the actual at-sea flight tests.

2.0 Description of Aircraft

2.1 The V-22 Osprey is a tilt-rotor, V/STOL multi-mission aircraft manufactured by Bell Helicopter Textron Incorporated and Boeing Helicopters. Each of the two 6,150 shaft horse power Allison (YT406-AD-400) turboshaft engines are housed in a wingtip nacelle and drives a 38-ft diameter, three bladed proprotor. The aircraft is a high wing design with retractable, tricycle type landing gear. The airframe is constructed primarily of graphite-reinforced epoxy composite material. The two proprotors are interconnected through a series of drive shafts and gearboxes to maintain rotor synchronization and one engine inoperative power to both proprotors. The tilt-rotor design allows the nacelles to rotate through a 97.5 deg arc, from horizontal (0 deg) in the fixed wing mode of operation to aft of the vertical (90 deg) in the helicopter mode of operation.

2.2 The design includes a triple redundant, fly-by-wire digital Primary Flight Control System (PFCS) and an Automatic Flight Control System (AFCS) to control the aircraft. The Flight Control System (FCS) software version used throughout this test was 5.3.2 (patch 064). This version encompasses the PFCS and core AFCS. With the nacelles at or near 90 deg (helicopter mode), the FCS controls the aircraft in a method similar to a twin rotor helicopter. During conversion of the nacelles to 0 deg (airplane mode), the FCS phases out the cyclic rotor swashplate commands and controls the aircraft through the use of conventional aerodynamic surfaces.

2.3 The test aircraft, BuNo's 163913 and 163914, were prototype aircraft which contained on-board flight test instrumentation. This instrumentation provided data from the test aircraft to both contractor and government personnel aboard the test ship. Additionally, aircraft 163913 was configured without infrared suppressors; in their place were installed prototype engine exhaust ejectors and 60 deg deflectors. These components were designed to deflect exhaust gases away from the aircraft and equipment aboard ship.

3.0 Description of Simulator

3.1 The U.S. Navy's V-22 Osprey simulation used for this series of tests was located in the NATC Manned Flight Simulator (MFS) facility. The computer models were hosted on the Tactical Avionics Simulation Test and Evaluation Facility computer system cluster. This system is a multi-configuration collection of Digital Equipment's VAX series of machines in association with aircraft mission and flight control computers, visual system generation devices, and the required bus and network interfaces.

3.2 The aerodynamic models were run on an Avalon Inc. processor which is hosted on a Digital MicroVax II. Another MicroVax II is used for the avionics system modeling. These models communicate with a V-22 configured AYK series mission computer via a 1553 bus patch panel. The mission computer in turn communicates to the aircraft quality display processors via 1553 bus interfaces in the simulator. The avionics, airframe and other processes share a common memory which resides in the in-house designed and constructed multiport memory.

3.3 The visual system used is a General Electric Compuscene IV, and the images produced by this system are projected on a Rediffusion Wide II visual display device. This display system is mounted to a Rediffusion six degree of freedom motion platform, which has been modified to accept multiple cockpits in keeping with the MFS's modular design. This combination motion base and visual projection system make up the motion simulation station.

3.4 The eight channels of digitized aural cueing are provided by a pair of Amiga computers. The sound samples themselves have been gathered from both the V-22 and the XV-15 tilt-rotor aircraft.

3.5 The cockpit itself was constructed to replicate the aircrew work stations of aircraft number BuNo 163912. It is a wheel mounted, moveable device that can function in any of the four simulation stations. For the training described in this paper the cockpit was utilized in the motion station described above. The cockpit is outfitted with actual aircraft Multi-Function Displays (MFDs), display processors, keypads and video display units, and high fidelity simulated instruments.

3.6 The ship models and the ship environment models have all been programmed in-house. State-of-the-art modeling techniques have been employed to replicate four different classes of U.S. Naval ships, and analytical updates are constantly investigated and tested to modify the characteristics of the ship environment.

4.0 Scope

4.1 Shipboard flight testing of the V-22 Osprey was conducted on 7 December 1990 and comprised 12 flight hours at-sea. The three government test pilots logged over 40 hours of simulator flight time during September of 1990 in preparation for the test mission. Engineering development and evaluations of the shipboard simulation began in June of 1990 and comprised over 165 hours of simulator flight time. Although simulation training and engineering analysis was conducted for the entire shipboard operations flight pattern, this paper will focus primarily on the most critical tasks: final approach, hover, and landing.

5.0 Measures of Effectiveness

5.1 Prior to the shipboard training, the V-22 simulator had undergone many tests of its ability to model the actual aircraft accurately in all flight regimes. The unique position of the Naval Air Test Center and the V-22 simulator in the development cycle of the aircraft has created a process that keeps the quality of the simulation at a relatively high state of flight dynamic accuracy at all times. The simulator is used to assist flight test planning of an upcoming portion of the project by optimizing the flight test profile, and also evaluate safety of flight concerns. Once the simulator has been checked out for training readiness by a member of the test team, the flight test cards are pre-flown in as realistic a manner as possible, with actual flight test participants performing the duties they would be assigned during the actual flight. These duties can include manning strip-charts, monitoring safety-of-flight data, or performing communications work. Part of the process is the establishment of all the parameters required for proper flight test data reduction, so a data base can be constructed of expected results by the time flight test occurs

5.2 After actual flight tests have been completed, the results can be directly compared to the predicted, and any required upgrades or corrections to the simulation are performed. Through this constant process of blending results back into predictions, the V-22 simulation has achieved an outstanding acceptance rate with the test pilots. The mission tasks have, qualitatively, been a comfortable and believable representation of the tasks the pilots perform in the aircraft. It was with this background that the aircrew entered training for the shipboard operations.

5.3 The most visible method of assessing effectiveness of simulator flying qualities and performance, and mechanical characteristics, is to quantitatively compare time history data of simulated flying with actual flight. Of course, if a certain aircraft configuration is planned to be tested with simulation prior to its incorporation into the actual aircraft, a 'best guess' of the predicted configurations flight characteristics must be made based upon prior simulation knowledge and experience. This was the case during the training and flight test preparation for the V-22's first shipboard testing and evaluation.

5.4 The use of a simulator prior to testing also provides a process for developing and fine tuning test methodology and procedures. Therefore, a post-test measure of the effectiveness of the simulator is to examine whether or not the methods and procedures developed during the simulator training sessions survived contact with the real world, or had to be modified during the actual flight test.

5.5 As previously stated, effectiveness can be measured both qualitatively and quantitatively. The following paragraphs in this section discuss the baseline criteria and expectations of the MFS and flight test engineers.

5.6 The aural cueing environment of the simulator should recreate noise levels and proper sounds, such as rotor sounds and wind noise, inside the simulator cabin. Warning tones and associated emergency sounds, such as an engine spooling down, should be representative of the aircraft.

5.7 The baseline flying qualities and performance aerodynamic model must be of sufficient fidelity as to not inspire contamination of data. Longitudinal, lateral-directional flying qualities, trimability, gust response, vibration, climb/descent level flight and hover performance, and engine characteristics must all be modelled to the criteria set forth by the engineers. This criteria is dependant on the level of influences each parameter is perceived to have on the given task.

5.8 The mechanical characteristics of the simulator must also be properly modelled. Variations in mechanical characteristics can influence both qualitative and qualitative test data. Depending on the task, or subtask, it may not be necessary to have a perfect or near perfect match of simulator and actual data.

5.9 The seat-of-the-pants 'feel' of the simulator should match that of the aircraft. This nebulous statement is the pilot's way of lumping together all the intangibles of the simulator and comparing them to the aircraft. The layout of the simulated cockpit, the feel of the seats, the control force/feel system, the windscreen glare, the field of view, in-flight body forces, and many other factors all add up to a single result for the aircrew: the simulator 'feels' like the aircraft or it does not. This is one of the hardest areas to quantitatively isolate in a simulator, and thus it can be one of hardest goals to accomplish.

5.10 The same feedback of actual data versus predicted results method is used with regards to the shipboard environment. This simulation is complex, encompassing the ship motion, ship airwake effects and the gear-ship deck interface. In this case, any known truth data is first gathered, then the simulation models are driven to match reality. For the first shipboard trials, limited data was available on ship motion and airwake, and the interaction of the aircraft with the deck was known only by prediction.

5.11 The shipboard environment should replicate that of the real event. The ship should be the same type, color, and size as the actual. The deck markings should be accurate, and the displacements of the ship island in relation to the deck landing spots should be visibly the same to the aircrew. Height and distance cues should be apparent, without need to reference instruments inside the aircraft for additional information. Ship bow and stern wakes, and the water surface must also not be forgotten. Most importantly, the textures and hues of the visual scene must effectively match the real world.

5.12 Finally, the pilots must personally feel they are getting effective training. What they are seeing must match both reality and their expectations sufficiently to inspire confidence in the ability of the simulator to prepare them for the mission.

6.0 Comparative Results

6.1 One method used for comparing the simulation to the actual event is through the estimation of power spectral densities (PSDs). The PSD is defined as the modulus-squared of Fourier transform, where the Fourier transform is given by

$$\mathcal{F}\{x(n)\} = \sum_{n=0}^{N-1} X(n)e^{k}$$

where

$$k = \frac{-i2\pi n}{N}$$

and the modulus-squared is given by

$$R_{x}(i) = \sum_{n=i}^{N} x(n)x^{*}(n)$$

where x* represents the complex conjugate of x, and so

$$\mathrm{PSD} = \mathcal{F}(R_{x})$$

6.2 In the case of the V-22 aircraft, this method is most useful for the lateral axis, the dominant workload axis for the aircraft. For the final approach (one-quarter mile to deck edge) section of the landing pattern, Figure 1 illustrates lateral stick activity, with simulator data presented as a solid line, and actual data as a dashed line.

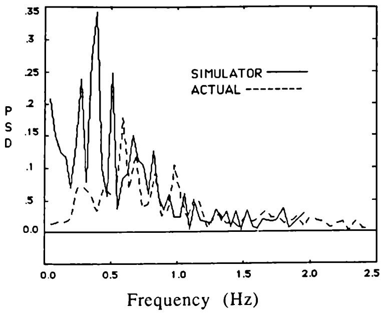


Figure 1
Final Approach

This data shows that during the approach to the deck edge, the actual event required lateral stick inputs of a higher frequency and lower power spectrum as compared to the simulation required to complete the same task. The actual task frequencies were centered around .52 Hz, and the simulation was centered around .48 Hz.

6.3 Figure 2 shows the lateral activity for the deck edge to touchdown task.

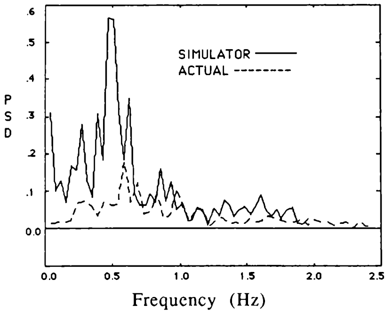


Figure 2
Hover to landing

It can be seen that during the landing phase the simulator required a higher workload than the actual event, and this workload occurred at a lower frequency. The simulator workload was over a wide range of frequencies, centered around .50 Hz, while the actual workload frequency remained in the .52 Hz range.

6.4 An important item to note from the stick activity presented is the percentage increase in workload from the first portion of the landing task to the final section. An increase of 27% is seen in the actual aircraft data, and an increase of 35% is shown in the simulator. Even though the overall workload in the simulator was higher than the aircraft, the increase in workload as the pilot transitioned from the near-ship environment to the landing environment was very similar. From a Test and Evaluation training standpoint, it is always better for the simulator to produce a more difficult situation rather as opposed to an easier situation for a given task. However, the differences must be documented and defined.

6.5 An analysis of flight control positions, power requirements, time requirements, and basic flight profile characteristics of the aircraft lends insight into how effective the simulation is and how well the update process works. Data are presented in Tables 1, 2, and 3 for three distinct tasks within the shipboard landing pattern; the final approach, the hover, and the landing. Zero percent indicates full left or full aft position, and positive indicates right bank angle.

Table 1
Approach Task

	Simulation	Actual
Rate of Descent (fpm)	250-270	200-225
Task Time Req'd (sec)	20-30	30-50
Lat stick (%max)/average	28-76 /50	48-55 /51
Long Stick (%max)/avg	10-34 /25	56-65 /59
Directional (%max)/avg	25-68 /50	46-49 /48
Mast Torque (%max)	20-83	25-85

Table 2
Hover Task

	Simulation	Actual
Hover Height (feet)	20	20
Task Time Req'd (sec)	8	10
Lat stick (%max)/average	38-57 /52	44-57 /50
Long Stick (%max)/avg	20-30 /25	51-55 /53
Directional (%max)/avg	46-58 /49	47-49 /48
Mast Torque (%max)	69-85	72-85
Bank Angle (deg)/avg	-3 to 4 /0	2 to 3/2.5

Table 3
Landing Task

	Simulation	Actual
Task Time Req'd (sec)	5 - 8	12 - 20
Lat stick (%max)/average	38-63 /52	39-59 /50
Long Stick (%max)/avg	20-30/ 25	50-56/ 53
Directional (%max)/avg	50-68 /56	48-52 /49
Mast Torque (%max)	85-74	85 - 76
Bank Angle (deg)/avg	-4to3 /-1	3 to 5/4

6.6 Controllability of the aircraft during approach is best evident by analysis of glide slope and closure rate with the ship. Initiation of final approach at similar airspeed, altitude and distance entry revealed excellent agreement between the actual task and the simulator. As expected, rate of descent was a bit shallower and closure rate slightly slower during the real-life event because of pilot caution.

6.7 For each task, with regard to control position and range, we see close agreement in lateral stick, directional, and power requirements between the simulator and the actual event. Only in the longitudinal axis do we see significant differences. Here the range of deflection is very similar, but the average position is approximately 25% different. This has been identified as a weight and balance modeling problem with the simulator caused by a incorrect assumption of aircraft configuration during the training load development.

6.8 The difference of 7 to 12 seconds in time required to land, with the longer duration during the actual landing, is attributed to the fact that actual tests revealed In Ground Effect (IGE) and Out of Ground Effect (OGE) phenomenon that required the pilot to stabilize at a lower hover prior to touchdown. The last anomaly seen is in comparison of aircraft bank angles while in hover and landing. During simulation flight, we see an oscillation of left and right bank as a result of the increased workload required in the lateral axis of the simulator, reduced visual cues, and insufficient modeling of rotor IGE/OGE effects. The actual task demonstrated constant right wing down in a hover which increased with decreasing altitude.

6.9 Aural cueing proved to be a system that gave good performance in the simulation. The aural cueing was not a specific factor that the pilots commented on, however, the absence of aural cueing in the simulation drew instant complaint. The sound levels actually experienced in the aircraft where higher and more distinct than the cueing in the simulator. The aural cueing seemed to provide a low level background feedback to the pilots, on which they did not always consciously key, but on which deviations and anomalies were instantly detected.

6.10 The control feel of the simulator was evaluated by comparing the mechanical characteristics of the simulator with the aircraft. Differences in mechanical characteristics can be a contributor to differences in pilot workload required for similar tasks. The test pilots indicated that the control mapping and control sense was indicative of the real aircraft, as were the characteristics of the power lever and pedals. Simulator cyclic trim system free-play was evident, yet control system free-play has yet to be determined. If the trim system free-play in the simulator exceeds the control system free-play, the pilot could easily be making control inputs within the deadband, which would, in turn, increase the workload.

6.11 The largest number of comments by the pilots were centered around the shipboard environment 'look and feel'. Altitude cues were not present in the simulation to the degree required, while during the actual evolution, altitude was easily and precisely determinable. The cause of this will be discussed later. Ship turbulence effects on the airframe were also lacking.

6.12 Aircraft procedures, such as pattern profile around the ship, very closely matched the actual requirements. This was perhaps the best simulated portion of the training, with one pilot commenting after his first landing "..I'd been here a thousand times before, it was just like the simulator."

6.13 Visual detail is usually a product of hardware processing capability. Exact field of view simulation would be the most desired, which would include not only forward and overhead windows, but also chin windows. The visual images themselves are simply a matter of programming time to prepare them for use, and processor capability. Presently the V-22 simulation at MFS lacks cockpit chin bubbles and overheads due to the roll-in roll-out configuration of the system, but presents state-of-the-art ship visual models, which are being constantly upgraded.

6.14 The two most significant problems discovered in the simulation during training were the airwake modeling in the near ship environment and the lack of visual cues in and around the shipboard region. For airwake modeling, two methods were examined during the course of the training. The first method attempted was a 'localized grid' method, in which a rectangular grid of ten by ten sections of airspace was mapped out to a range of one mile from the center of the ship deck. Each section was then assigned an up, down, and sideward 'base' air disturbance speed, and an acceptable plus-or-minus random component. When the aircraft center of gravity entered a region of turbulence, gust values were calculated for the block region by adding or subtracting the random component from the base values and then scaling the values based on altitude above the ship deck. These gust values were then included in the aircraft aerodynamic equations.

6.15 Transitions between grid frames were accomplished using a simple linear filter for each gust axis. Airflow velocities in each section were based on pilot experience, the limited flow visualizations that were available, and data gathered by the Dynamic Interface Department at the NAVAIRWARCENACDIV. In practice, this modeling methodology did not provide an appropriate frequency and amplitude of aircraft responses, and the model was abandoned during the simulator training effectiveness evaluation phase prior to the onset of training. The basic reasoning behind the model continues to be examined.

6.16 Another, simpler method of airwake modeling was then used. This method simply raised the pilot workload by placing low frequency white noise into the control system inputs as a function of position relative to the landing spot. This computer generated disturbance raised the required pilot workload to the level desired. This model noise was filtered out as the simulated aircraft transitioned over the deck edge, as the normal V-22 ground effect models were ramped in. This methodology caused several problems during the training and its utility is marginal at best. The problems encountered in this training have prompted several lines of study into bettering airwake models for the shipboard environment through various resources.

6.17 Shipboard visual cueing proved to be a significant problem area during training as well. The ship model used was upgraded in-house from the basic model provided with the visual generation system. The lack of a good sea surface and ship bow and stern wake models was found to be a problem early in the simulator evaluations, but no changes were implemented before training began. The ship model itself gave good visual cues during the final approach to hover portion of the pattern; but when the pilot's view was confined to the side of the ship superstructure and the flight deck during the final hover phase, the visual cues became less informative. The same problem was encountered during the extended portions of the flight pattern.

6.18 The sea surface model failed to provide adequate height cues to the aircrew and, even with 'ship' in sight, the pilot was forced to have the copilot call altitudes or frequently scan inside the cockpit to read altitude. The addition of aircraft support trucks and support equipment models on the flight deck surface helped provide cueing in the low hovers, but significant improvement remains to be made in this area.

6.19 The addition of ship water wake modeling and bow wave modeling to the visual scene provided a small measure of improvement during flight over water. Ship wakes helped provide the pilot with critical lineup cues. The Navy is improving the ship visuals by producing a very high detail amphibious assault ship model for further training which will incorporate many details of the ship, such as external walkways and ducting on the side of the ship superstructure, hatch-covers, and other features on the deck. A solution for the sea surface problem is still being investigated.

6.20 The six degree-of-freedom ship motion model was derived from actual test data. This model, when plotted against actual recorded ship motions, compared very well. During the training sessions, however, the pilots had trouble perceiving this projected motion. The ship motion model had gains ranging from 1.5 to 3.0 placed on the six channels of output to boost the apparent motion. This perception problem appears to stem from the lack of visual cues on and around the flight deck area and the simple visual sea surface model, coupled with the training not taking place under platform motion. The ship motion model should serve very well once its precise outputs are no longer lost in the perceptual noise of the other cueing devices, such as the gear model.

6.21 The aircraft aerodynamic model and rotor model performed well for the mission training. The lack of a blade element model in the simulation reduces the simulators response to a ship airwake model.

6.22 IGE and OGE aerodynamic models incorporated in the basic airframe model give good results for land-based training, but lack the sophistication required to simulate the one rotor IGE/one rotor OGE situation found during shipboard operations. Data from the first ship trials is enabling the creation of such a model for the simulation training that will precede the next series of sea trials.

6.23 The landing gear model used in the V-22 simulation is an adaptation of the generic gear model used in all of the simulations, ranging from AV-8 to F/A-18 to the V-22 aircraft. Using this gear model in a slow vertical landing onto a six degree-of-freedom deck regime was one of the biggest challenges in our training preparation. The breakout frictions required to keep the aircraft from sliding about the deck at idle power were noticeable to the pilots. Upon application of power by the pilot to taxi the aircraft, the aircraft would not move at all at low power settings. When a power setting 10-20% higher than that required in the real aircraft was reached, the simulation would 'breakout' and roll with a perceptible jerk. Also, when landing on spots at the extreme stern of the ship, the gear damping would produce a heave motion 180 degrees out of phase with the ship motion. Touchdown cues were not apparent in the current model due principally to the lack of motion base usage.

6.24 Hardware also played a large role in the less effective areas of the training. The motion base was not used during this training. The motion algorithms developed in-house for the V-22 at the time were not fully verified, and the degree of confidence in them was low. Since the training has been completed, these algorithms have been refined and they have been used in subsequent training sessions. It is believed that use of the motion base will help the pilot perceive ship motion and airwake turbulence through body forces and provide helpful cues for hover work around the ship.

6.25 The simulation visual display system was hampered by the lack of a view through the chin windows of the aircraft. The V-22 simulator cockpit is equipped with these windows, but the confines of the motion base, and the requirement to keep the motion base usable for all the MFS simulations precludes the mounting of visual display devices for these windows. The lack of these windows, coupled with the low visual cues from the flight deck, created a higher workload situation during low hovers and landings in the simulation.

6.26 Computer processor time, in the form of time delays, also played a part in the training effectiveness of the simulator. The measured time delay in the system, from pilot stick step input to visual system update, was found to average 128 milliseconds greater than that experienced in the actual aircraft. The technology is in place today at the MFS to reduce the average time delay to 95 milliseconds with new computer hardware and modeling techniques. This would further improve the training effectiveness of the device by pushing simulator responses more toward the response times of the aircraft.

7.0 Conclusions

7.1 The overall training effectiveness of the simulator was qualitatively ranked by the test pilots as excellent for the task. Most of the deficiencies made the simulator training tasks harder to perform than the actual event, thus avoiding instilling any false sense of security.

7.2 Pilots noted that the simulator, through task repetition, instills task discipline and familiarity and increases safety. The simulator, though lacking in some areas, gave the pilots an excellent first impression of the task. This early impression reduced the pilot's anxiety of conducting tasks in the aircraft, thereby reducing stress levels and improving pilot concentration and efficiency.

7.3 In terms of training, test pilots also commented that the ability to practice shipboard emergency situations, and the proper corresponding emergency procedures was invaluable. Some of the emergency situations trained for were : run-on landings, one engine inoperative landings, and wave-offs. Two of the situations trained for, the case of one gear failing to extend and wave-offs, were encountered during the test period and the pilots reacted as trained without further incident.

7.4 When a pilot flies a new aircraft type aboard ship for the first time, his initial landings and takeoffs normally display noticeable aircraft attitude, control, and power deviations in frequency and/or in deflection during the approach profile, hover, and landing phases until the pilot feels comfortable with the aircraft and the shipboard environment. As revealed by time history data, in Enclosure 1, these type of deviations were hardly evident during initial landings and takeoffs during the V-22's initial shipboard trials due to effective pilot training.

7.5 Simulator preparation enabled the test team to define and perfect the 'control strategy' the pilots would use to fly the tilt-rotor aboard the ship. Because of the unique acceleration and deceleration capability of the tilt-rotor, pilots had to develop new longitudinal control habits using articulating nacelle control. This habit pattern was developed in the simulator and verified during actual training flights. Without the benefits of a high fidelity simulation, flight training costs could easily have increased tenfold, with an attendant increase in risk.

7.6 We believe that use of the simulator enhanced safety of flight and no mishaps occurred during the actual test. This was a hazardous test in a prototype aircraft under demanding shipboard conditions. The fact that our test pilots conducted all launches and recoveries with precision can be attributed to a large degree to the effective simulator training.

8.0 Future Plans

8.1 As the developmental test and evaluation schedule of the aircraft progresses, there will once again be training at the Manned Flight Simulator in the shipboard environment for the next evolution at sea. For support of this effort, work focuses on several areas in the time allowed to improve the performance of the system.

8.2 The largest improvement will be found in the visual modeling area. A new amphibious assault ship model is being prepared which will contain eight times the number of polygons found in the previous model. This extra detail will be concentrated in the areas that the pilots have found need the most work, such as the side of the superstructure and the flight deck edge. An animated Landing Signalman Enlisted figure will be added. In addition, this model will be Night Vision Goggle compatible to support a broad spectrum of testing conditions. It is hoped that the visual model improvements will make the ship motion more apparent and use of the realistic, unaugmented ship motion model will become possible.

8.3 The landing gear model will be further tuned to give more realistic breakout and friction values as data becomes available. If the apparent ship motion becomes more readily detectable by the pilots, the more realistic ship motions will require less compensation in the gear modeling. The use of the motion base will give touchdown cues and strut and tire dynamic cues lacking in previous training.

8.4 The motion base will also be employed for this series of training. The algorithms were used successfully in the most recently completed training sessions. The increased cueing should upgrade the training effectiveness.

8.5 The V-22 simulation continues to provide excellent flight test support through constant upgrading and effort on the part of U.S. Navy personnel. This simulation capability is being employed in other applications not originally planned; such as cockpit lighting studies, crew coordination studies, hardware reliability and maintainability studies. In addition the modular code designed for the trainer has been reused in a deployable aircrew coordination trainer and has provided data for the Operational Flight Trainers now under development.

8.6 By beginning the simulation effort very early in the aircraft life cycle and by constant upgrades to maintain configuration with the evolving aircraft, the NAVAIRWARCENACDIV MFS has produced an effective and proven flight test support tool.

9.0 Acknowledgements

9.1 The authors of the paper would like to note that the simulation described above is the work of the many dedicated professionals at the Manned Flight Simulator facility. The following specific people should be recognized for their roles in the simulation training being discussed above : Mr. Danny Campbell, Mr. Joe Kelponis and Mr. Mike Hughes. Mr. Jeff Weathers deserves special recognition for his work with the V-22 flight control laws.

9.2 Additionally, the leadership and foresight of the Naval Air Systems Command V-22 Program Manager, Colonel J. Schaefer (USMC), and the Assistant Program Manager, Major R. Curtis (USMC) in establishing and maintaining this simulation capability has been instrumental for the flight test support of the MV-22 Osprey.

AIAA-92-4131-CP

Simulation of Automatic Rotorcraft Nap–of–the–Earth Flight in Graphics Workstation Environment

T. Lam
Sterling Software
Palo Alto, California

Victor H.L. Cheng
NASA Ames Research Center
Moffett Field, California

Abstract

This paper describes a 3-dimensional (3D) helicopter flight simulation system. The simulation is designed to be a readily available tool for concept verification and tuning of automatic obstacle-avoidance guidance algorithms. The system has been implemented on networked workstations capable of interactive 3D graphics simulation. The simulation uses realistic terrain and obstacle models. The dynamics of the rotorcraft and the functional capabilities of the range sensors are simulated to provide all the components required to evaluate the guidance function. Standard graphics hardware available on the workstation is utilized to accelerate the range-data calculations for sensor simulation at the guidance rate. An example is given to demonstrate the performance of the obstacle-avoidance capability.

1 Introduction

In high-threat battle areas, combat rotorcraft fly at low altitudes to minimize the risk of being detected by the enemy. There are three common modes of low-altitude flight that the U.S. Army categorize as terrain flight. They are the low-level, contour and nap-of-the-earth (NOE) modes, and maximum concealment is provided by NOE flight. In the NOE mode, helicopters fly as close to the earth's surface as vegetation and obstacles will permit. A typical scenario may involve the helicopter flying below tree tops, moving at the highest possible speed while following contours of the terrain. The vegetation which serves as the means of concealment also poses as hazardous obstacles to the helicopter. An automatic guidance system which can successfully perform the piloting function for NOE flight will enable the pilot to focus more attention on accomplishing mission objectives than on obstacle-avoidance maneuvers. The Automated NOE Flight Program was initiated at NASA Ames Research Center to research the feasibility and initiate development of technologies for automating NOE flight to reduce pilot workload [1,2].

To adequately address the constant danger of rotor strike in NOE flight, the verification and validation of automatic obstacle-avoidance guidance concepts can benefit substantially from simulation tools that are flexible enough for easy modeling of different flight scenarios. The simulation tool employs 3D graphics to depict helicopter, terrain, and objects such as trees, houses, and telephone poles to represent different kinds of obstacles. Such 3D models facilitate the visualization of the helicopter flight in real time with due regard given to the physical dimensions of the vehicle and the obstacles.

Full-scale simulations which use high-order dynamical models, detailed terrain scenery and real-time graphics together with motion feedback are useful for pilot-in-the-loop evaluation. However, the complexity and cost of these facilities limit their availability and therefore prohibit their wide use. Our objective is to develop a readily available interactive simulation tool which can be manipulated easily to explore new ideas in helicopter guidance automation. With the availability of relatively inexpensive workstations with high 3D graphics performance, less-complicated simulations can be performed for the purpose of concept verification, as well as for module testing during the development phase.

This paper describes a 3D helicopter flight simulation system for evaluating automatic obstacle-avoidance techniques. The system has been implemented on general purpose graphics workstations. The system provides texture-mapped terrain and obstacle models for realistic simulation. The dynamics of the rotorcraft and the functional capabilities of the required range sensors are simulated to provide all the components for guidance function evaluation. Graphics hardware available on the workstation is utilized to accelerate the range-data calculations for sensor simulation fast enough to keep up with the guidance rate. Section 2 of this paper contains the description of all components of the simulation system. Section 3 discusses speed performance issues of the implemented system. Examples of helicopter NOE flight

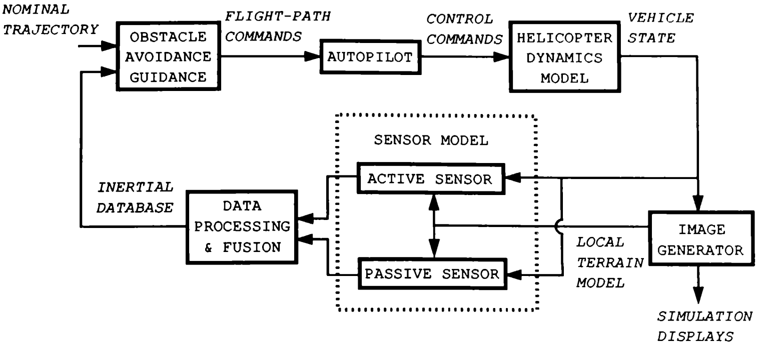


Figure 1: Structure of Automatic Obstacle–Avoidance Guidance Simulation

assisted by the automatic guidance system are presented in section 4. Section 5 concludes the paper with suggestions for future work.

2 System Modules

Automatic obstacle avoidance requires precise obstacle information. Information related to the locations and sizes of all the obstacles in the rotorcraft's surrounding environment is needed to assure flight safety. It is, however, unlikely that the pilot will have all the obstacle information available to the guidance system in the mission-planning stage, and so such information must be collected in flight in real time. The problem can be resolved by installing an onboard obstacle-detection system which collects real-time obstacle information together with sensed terrain data. The collected and processed data can then be passed to the obstacle-avoidance guidance module to adjust the flight path from the originally planned nominal trajectory.

Aspects involved in the automatic guidance system can be explored by building a simulation tool with the following components:

- Helicopter Dynamic Model
- Image Generator
- Sensor Model
- Sensor Data Processing
- Obstacle Avoidance Guidance
- Autopilot

Figure 1 illustrates the relationship between these various subsystems of the simulation system.

2.1 Helicopter Dynamic Model

The helicopter is modeled as a rigid–body system with six degrees of freedom. The aerodynamics is specified by stability derivatives corresponding to a generic helicopter plus a first-order engine lag response. It uses a simple closed-form trim solution, and linearized quasi-static aerodynamic force and moment equations. The dynamic model calculates the forces and moments on the aircraft and imposes some constraints on the vehicle response and maneuverability. The inputs to the dynamics model include longitudinal and lateral cyclics, collective and rudder settings. Detail of the dynamics model is described in [3].

2.2 Image Generator

Helicopter flight has traditionally been difficult to simulate since it flies at a wide range of altitudes with different display requirements. The image generator is required to be able to display a large area of terrain for high-altitude flight and to handle ground detail in NOE flight. These requirements are further complicated by the limited computing resources available from the general purpose graphics workstation on which the simulation is implemented.

The Silicon Graphics 4D VGX graphics workstation is selected for the implementation based on its price, general computing and 3D graphics performance. The workstation has four MIPS R3000 processors running at 25 MHZ, 32 Mb of main memory, and dedicated graphics hardware support for texture-mapping, lighting and depth-buffering.

Due to the limited memory resources, the image generator maintains a simplified terrain and object database for fast data traversal. The locations and attributes of terrain and objects are organized and stored in rectangular grids. The system executes an automatic culling process which selects and traverses grids within the current field-of-view and up to a specified range selectable during run time (Figure 2). Database traversal is executed in such a way that vertices which are referenced multiple times within an object are specified only once. Reductions in vertex specification provide a direct reduction in data transfer requirements and rendering calculations. For example, terrain grids are rendered as triangular meshes where common vertices between neighboring polygons are shared.

Since NOE flight is in close proximity to the terrain and obstacles, emphasis is placed on terrain and obstacle modeling details. The size and variation of texture data is limited by the 256 × 256 × 32 chunk of hardware texture memory available on the workstation. This limit is strictly adhered to since data swapping from the texture memory will serverly penalize the graphics rendering rate. The terrain is covered with a 128 × 128 texture pattern repeating over the entire visible region. Trees are modeled with polygons textured with various tree images, which include two levels of detail available for selection based on their range from the simulation-display viewpoint.

The simulation provides the capability of scenario display from three different perspectives simultaneously (Figure 3). The cockpit view shows terrain as seen from the pilot view point with the head-up-display symbols superimposed over the scene. The bird's-eye view displays the helicopter along with its surrounding terrain area and its view point location is interactively adjustable. The wingman view simulates views as seen from another helicopter that is closely following the first. System features such as this multi-viewing and other capabilities are controlled through a user interface panel which consists of widgets such as buttons, dials, sliders. The system also allows interactive adjustment and monitoring of selected parameters in the program.

The terrain database accepts data stored in Digital Elevation Model (DEM) format produced by the U.S. Geological Survey. The DEM consists of a sampled array of elevations for a number of ground positions and is available at several different regularly spaced intervals.

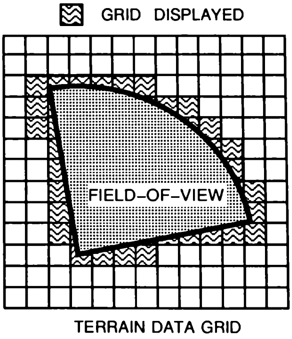


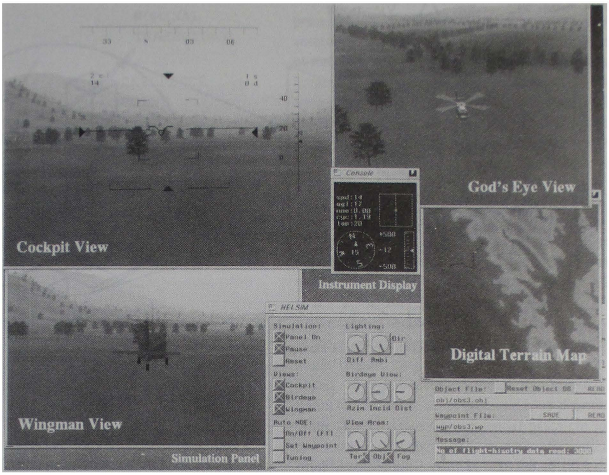
Figure 2: Terrain Grid Data Travisal

Creating and modifying obstacles in the scenario are achieved through an interactive 3D terrain-object editor (Figure 4). The editor allows the user to visualize the terrain from different adjustable perspectives. Types of objects available for modeling include trees, houses, walls, telephone poles and wires. In addition to some of the standard editing operation such as addition, deletion, copying and scaling, the editor also provides convenient functions such as random object creation and automatic object alignment.

2.3 Sensor Model

Current research efforts indicate that the ranging requirements of obstacle-detection systems for automatic guidance will likely be satisfied using both active and passive sensors [4]. Active sensors include millimeter-wave radars and laser systems capable of providing range measurement by radiating electromagnetic energy. Range estimates can also be extracted from passive sensors such as forward-looking infrared or low-light-level television cameras using computer-vision techniques.

Our obstacle-detection research emphasizes the use of passive sensors because of their potential wide field of view and covert nature. Since passive sensors do not emit detectable radiations, they are especially preferred in the hostile environment that require NOE flight. The concept of reconstructing 3D spatial information from images through stereoscopic considerations is described in [5-8]. The apparent shift of objects in successive image frames together with the given aircraft motion can be used to infer range to objects within the field of view. Current implementation of



spd:14
agl:17
noe:0.00
cyc:1.19
tem:20

+500
-12
-500

Simulation:
☒ Panel On
☒ Pause
□ Reset

Views:
☒ Cockpit
☒ Birdeye
☒ Wingman

Auto NOE:
□ On/Off (F1)
□ Set Waypoint
□ Tuning

Lighting: Diff Ambi Dir □

Birdeye View: Azim Incid Dist

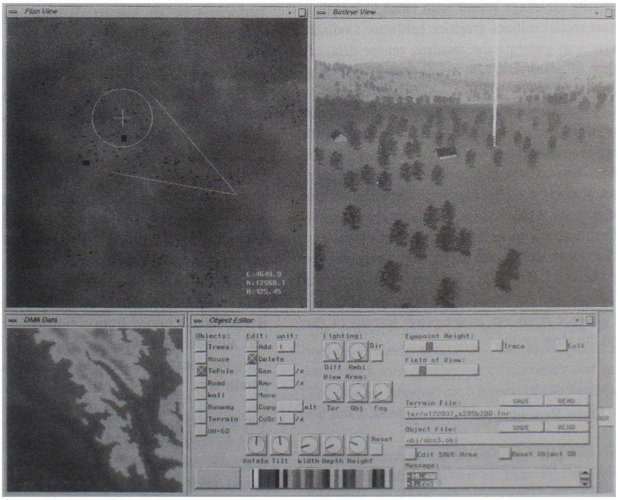
View Area: Ter ☒ Obj ☒ Fog

Object File: □ Reset Object DB READ
obj/obs3.obj

Waypoint File: SAVE READ
wyp/obs3.wp

Message:
No of flight-history data read: 3000

Figure 3: Simulation Displays with user control panel



L:4643.9
N:12968.1
A:425.45

Objects:
□ Trees:
□ House
☒ TePole
□ Road
□ Wall
□ Runway
□ Terrain
□ UH-60

Edit: unit:
□ Add 1
☒ Delete
□ Gen /s
□ Rmv /s
□ Move
□ Copy alt
□ CoSc 1 /s

Lighting: Diff Ambi Dir □

View Area: Ter Obj Fog

Rotate Tilt Width Depth Height Reset □

Eyepoint Height: □ Trace □ Exit

Field of View:

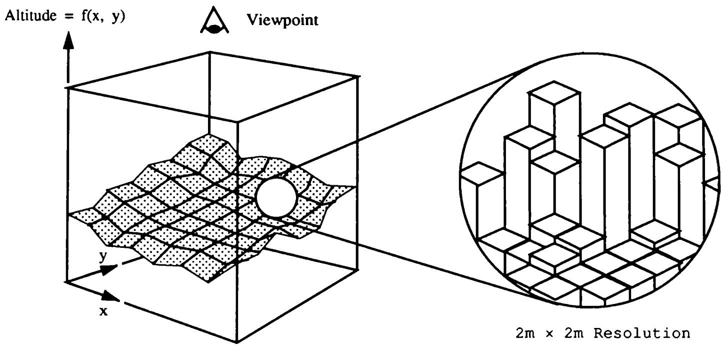
Terrain File: SAVE READ
ter/u122037_s285b200.ter

Object File: SAVE READ
obj/obs3.obj

□ Edit SAVE Area □ Reset Object DB

Message:
-19.400
-1.7

Figure 4: 3D Terrain-Object Editor



3D Orthographic Projection for Data Fusion

Figure 5: Inertial Database Model

the simulated sensor assumes range estimation can be obtained for every pixel within the sensor field of view, with no specific assumption on whether the sensor is active or passive.

In order to simulate the range sensor with close to real-time speed, the sensor model utilizes graphics hardware available on the workstation to generate the required sensor data. A graphics window of pixel size 128 × 128 (assumed size of sensor) is set up in the frame buffer. Simulated scenes as viewed from a forward-looking sensor mounted at the nose of the helicopter are transformed, projected, depth-buffered and scan-converted into the window. The depth-buffer which stores the distance from the image plane to each pixel within the field of view contains the range data that we are seeking. Thanks to the system design which allows access to the frame buffer, data stored in the depth-buffer planes can be retrieved. To obtain the precise range estimates given the potential wide field of view, the data retrieved from the depth-buffer has to be converted since it represents the perpendicular distance from the image plane instead of the true range from the view point.

2.4 Sensor Data Processing

As discussed in [4], state-of-the-art computer-vision techniques may not be able to provide complete range information within the sensor's field of view. Future implementations will address the problem where only sparse range data are available from passive sensor. Under this condition, supplementary data can be obtained by minimal use of active sensors in an efficient manner based on information previously obtained by passive means. Such a multi-sensor system necessitates fusion of sensor data, especially if two sensor systems are not being updated simultaneously. Sensor data fusion is most conveniently performed in an inertial database expressed in Cartesian coordinates.

The inertial terrain and obstacle database stores altitude information over a 700m × 700m area surrounding the rotorcraft's position. The database is organized as a square grid divided into 7 × 7 square subgrids, with each subgrid of size 100m × 100m. The grid has a horizontal plane resolution of 2m × 2m. The helicopter is always positioned within the center subgrid of the database. As the helicopter moves out of the center subgrid, the database is shifted by 100m accordingly so that the rotorcraft remains within the center subgrid. During this process, portion of data is shifted out of the database to make room for the new area which is first initialized with interpolated DEM altitudes. Range data obtained from simulated sensors are transformed and altitude information is stored. In this implementation, only the maximum altitude is saved as a function of the terrain area.

The graphics hardware of the workstation is utilized again in order to accelerate the inertial database calculations. The 7 × 7 square grid terrain contained in the inertial database is placed inside a 3D orthographic projection enclosure with the view point directly above the center of the database and view direction downwards (Figure 5). The far and near clipping planes of the view volume correspond to the zero terrain datum and an arbitrary maximum elevation. The database

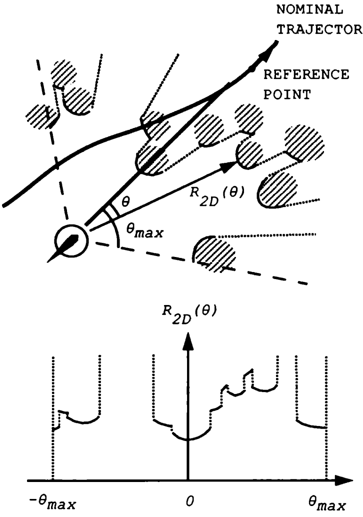


Figure 6: 2D Range Map Concept

(depth-buffer) is first initialized when terrain polygons are scan-converted into the frame buffer. The range information of the simulated sensor is then transformed and mapped into the view volume according to the geometric orientation of the helicopter. During this mapping process, the depth-buffer hardware automatically compares the elevation value associate with each of the sensor pixels against the altitude previously stored at the corresponding terrain cell position. The comparison stores the maximum altitude in the depth-buffer. Inertial-database data shifting due to helicopter position update is accomplished with a graphics utility function that copies pixels from a region of the depth-buffer to another region.

2.5 Guidance Algorithms

The goal of the guidance system is to follow a pre-determined nominal trajectory and divert from it only to avoid collision with obstacles detected by sensors. The nominal trajectory is a flight path determined by a higher-level guidance module [9] or provided as pilot commands [10].

The sensor information gathered in the inertial database is presented to the guidance module to generate flight-path control commands. This involves the following functions: obstacle/terrain discrimination, two-dimensional (2D) range-map computation and path selection, altitude-profile computation and 3D flight-path definition. This particular guidance design emphasizes the 2D flight path selection by decoupling it from altitude tracking because helicopters in NOE flight prefer flying around obstacles instead of flying over them.

The obstacle/terrain discrimination function provides a planar view of the obstacle situation based on the inertial database by identifying large local altitude variations. The obstacle information is then used to generate a 2D range map over a sector, generally larger than the sensor's instantaneous field of view, in front of the vehicle. This range map provides range estimates as a function of azimuth angle (Figure 6). A reference point located at a distance ahead of the helicopter along the nominal trajectory is defined. It serves as an instantaneous directional reference. With the 2D range map, potential paths are identified based on the concept of range discontinuities. A minimum-deviation path is selected to direct the vehicle to move along the nominal trajectory provided by the higher-level guidance. The altitude profile along this 2D path is determined based on the inertial database, and it dictates the vertical flight-path command.

2.6 Autopilot

With the vehicle heading vector determined by the guidance module, the autopilot calculates and adjusts helicopter control settings which include cyclics, collective and rudder. The current implementation makes use of a nonlinear functional inverse of the helicopter dynamics to provide the control inputs [11].

3 Performance Considerations

The implemented simulation system has been tested with various obstacle scenarios. The speed performance depends on factors such as number of obstacles and graphics details. The timing information for each of the modules presented in section 2 is shown in Table 1.

Within the simulation loop, the most time-consuming functions are the image generator and sensor data processing modules. However, the sensor data processing is performed only at every guidance cycle, which is executed once for every 10 simulation cycles. The workstation graphics hardware is able to substain an update rate of approximately 10 Hz with the minimum graphics details required for the obstacle-avoidance and guidance evaluation. We are interested in solutions that can retain the desired properties of real-time helicopter dynamics, and yield on overall motion smoothness. Based on these observations, the simulation system is further improved by techniques described as followed.

The helicopter dynamic model is simulated at 20 Hz cycles. However, the dynamic model's average computer run time is much less than its integration time-step increment (0.05 sec). This lead time can be used to compensate for the relatively slow graphics display process. By suppressing the graphics displays of the simulation cycle whenever the dynamic model clock is slower than the real-time clock, real-time dynamics behavior of the helicopter is preserved.

As discussed before, the use of graphics hardware for sensor simulation and sensor fusion was intended to accelerate the range calculation and thus the system's performance. However, this sharing of the single graphics pipeline on the workstation also restricts the image generator from achieving its optimum performance: the image generator must halt its operation when the graphics pipeline is occupied by the sensor modules. An improved version of the simulation was implemented over two networked graphics workstations. In this implementation, the autopilot, helicopter dynamics model and image generator are processed on one workstation while the sensor and guidance modules are executed on the second. This distributed configuration allows both processes to maintain their own graphics resources and run in parallel. Furthermore, the task of communication between processes over the network is being handled by another cpu on the multi-processor system and, therefore, the interruption on simulation process is minimized.

4 Examples

The simulation was tested in different obstacle scenarios with forests of varying density and arrangement. The following examples illustrate how the helicopter navigates through a terrain scenario in NOE mode with the aid of the automatic guidance system.

As discussed earlier in the paper, the nominal trajectory is provided by a higher-level guidance module. In this implementation, the nominal trajectory is defined by piecewise linear segments on the terrain database. The nominal trajectory is represented by the white lines in Figure 7. The retrace of flight history positions shows that the helicopter flies in close proximity to the pre-defined nominal trajectory and avoids obstacles detected by the guidance system while flying below the tree tops.

System Module	Average Time (seconds)
Sensor Simulation	.04
Sensor Data Processing	.12
Guidance Algorithms	.09
Autopilot and Helicopter Dynamics Model	less than .01
Image Generation	.10

Table 1: Average processing time breakdown for each simulation modules.

5 Conclusion

This paper covers the implementation of an obstacle-avoidance guidance simulation system on general purpose graphics workstations. The system consists of all essential components for guidance function evaluation. With the realistic 3D graphics model, high speed performance and interactive control interface, the simulation has proven to be an effective development tool for the guidance research effort.

Although the current general-purpose graphics workstation lacks the capability for real-time graphics support, its generality and affordability allows users to create customized solutions at minimal cost. Their networking capability allows distributed computing and multi-stationed simulation. In fact, the simulation has recently been modified for evaluation of different concepts of pilot interaction with the automatic obstacle-avoidance system, where as many as three workstations have been networked to share the computation.

With future advances in graphics technology and hardware miniaturization, "embedded simulation" where image generator driven by on-board avionics will permit the aircraft to become the ultimate simulator.

References

[1] V.H.L. Cheng, and B. Sridhar, "Considerations for Automated Nap-of-the-Earth Flight," *AHS Journal*, Vol. 36, No. 2, pp. 61-69, April 1991; also, *Proceedings of the 1988 American Control Conference*, Atlanta, GA, June 15-17, 1988.

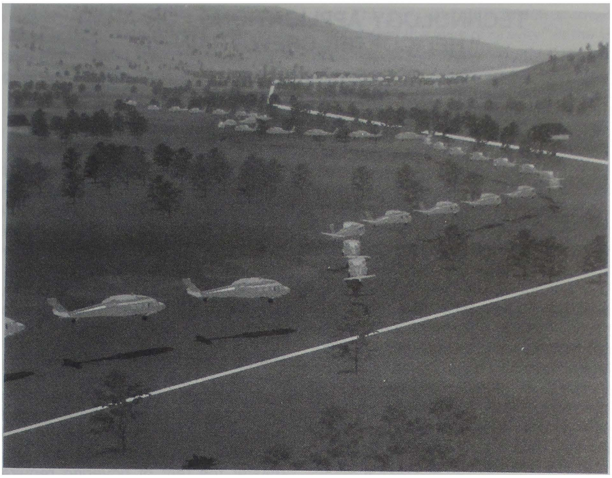


Figure 7: Example of Automatically Guided Flight Path

[2] V.H.L. Cheng, and B. Sridhar, "Technologies for Automating Rotorcraft Nap-of-the-Earth Flight," *Proceedings of the 48th AHS Annual Forum*, June 3-5, 1992.

[3] M.S. Lewis, and E.W. Aiken, "Piloted Simulation of One-on-One Helicopter Air Combat at NOE Flight Levels," NASA TM 86686, April 1985.

[4] V.H.L. Cheng, and B. Sridhar, "Integration of Active and Passive Sensors for Obstacle Avoidance," *IEEE Control Systems Magazine*, Vol. 10, No. 4, pp.43-50, June 1990; also, *Proceedings of the 1989 American Control Conference*, Pittsburgh, Pennsylvania, June 21-23, 1989.

[5] B. Sridhar, and V.H.L. Cheng, "Computer Vision Techniques for Rotorcraft Low-Altitude Flight," *IEEE Control Systems Magazine*, June 1988, pp. 59-61.

[6] B. Sridhar, and A.V. Phatak, "Simulation and Analysis of Image-Based Navigation System for Rotorcraft Low-Altitude Flight," *Proceedings of the AHS Conference on Automation Applications of Rotorcraft*, GA, April 1988.

[7] B. Sridhar, V.H.L. Cheng, and A.V. Phatak, "Kalman Filter Based Range Estimation for Autonomous Navigation Using Imaging Sensors," *Proceedings of the 11th IFAC Symposium on Automatic Control in Aerospace*, Tsukuba, Japan, July 1987.

[8] B. Sridhar, R. Suorsa, and B. Hussien, "Passive Range Estimation for Rotorcraft Low Altitude Flight," NASA TM 103897, October 1991; also to appear, *International Journal of Machine Vision and Applications*.

[9] V.H.L. Cheng, "Concept Development of Automatic Guidance for Rotorcraft Obstacle Avoidance," *IEEE Trans. on Robotics and Automation*, Vol. 6, No. 2, April 1990, pp. 252-257.

[10] R.A. Coppenbarger, and V.H.L. Cheng, "Concepts for Pilot Interaction with an Automated NOE Obstacle-Avoidance System," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Hilton Head, SC, August 10-12, 1992.

[11] V.H.L. Cheng, and T. Lam, "Automatic Guidance and Control Laws for Helicopter Obstacle Avoidance," *Proceeding of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France, May 10-15, 1992, pp. 252-260.

AIAA-92-4132-CP

TECHNOLOGY APPLICATIONS FOR ARMY HELICOPTER CREW TRAINING

Samuel Knight
Robert Monette
CAE-Link Corporation
Binghamton, New York

Abstract

This paper discusses the requirements for helicopter crew training and how the U.S. Army systematically approaches developing crew skills. It explains how basics are taught which lead to individual crew member proficiencies, as well as methodologies to develop crew coordination and tactical decision making. Also addressed is the hierarchy of training devices employed by the Army to develop crew skills. This includes discussions on how simulation technologies have been applied to meet the training requirements of each hierarchical level. As an example, we discuss the training devices used in the AH-64 Apache transition and how these task, procedural, and full fidelity devices are used in a building block approach to transition crew members from basic flight to flying and fighting in a hostile environment and, subsequently, to develop them into an integrated crew that can execute exacting crew and team decisions in a tactical battle scenario. This paper also delves into current trends to further enhance crew combat readiness through team and combined arms training. Also discussed are the emerging requirements for mobile training devices.

Introduction

The basic requirements for employing simulation technologies to train military helicopter crews are well recognized. Safety, cost, and environmental concerns are key factors. The training goal is to produce combat-ready crews; crews that can employ complex navigation and weapons systems while flying low-level, nap-of-the-earth profiles in a hostile environment, often at high speeds and often at night. The challenge is to provide this training with minimal physical risk to the students, with minimal expense in terms of operations costs and combat assets (vehicles, fuel, weapons, etc.), and without risking environmental damage.

Although the emphasis of this paper applies to crew training, it must be understood that pilots must possess individual skills before they can be absorbed into an integrated crew. Students must not only learn skills, but must also learn when and how to apply them. Training devices have proven to be a valuable asset in improving and accelerating the learning process of complex tasks. Because of this, the requirement for varied levels of training devices continues to expand. At the U.S. Army Aviation Training Center at Fort Rucker, Alabama, numerous devices support a large percentage of the total curriculum in a number of advanced courses. These devices employ a wide range of simulation technologies.

Hierarchical Training

A student learning the skill of hovering in a real helicopter realizes that there are three basic requirements: the desire to succeed, a patient instructor pilot, and, most important, plenty of open space. To master the feat of hovering requires a coordinated application of the collective, cyclic, and pedals while only inches off the ground. This is not a simple task. The new age "Nintendo Grads" may possess a slight advantage over the older "Erector Set" generation; regardless, hovering is a sizeable challenge to master. Practicing in a building-block process is the only method to correctly approach this and the multitude of systems and crew coordination skills that are required to subsequently prepare a combat-ready crew. The building block methodology in a hierarchical fashion adds advancing degrees of difficulty as the individual has mastered certain skills and is prepared to face new and more complex challenges. As each level is mastered, it becomes the foundation upon which to support additional skill acquisition. These skills, however, are not like those required to ride a bike, i.e., once you know how to do it you never forget. On the contrary, the complex interactive skills aviators must perform to effectively maneuver a helicopter around a battlefield are a perishable resource. Hence, there is also a need for sustainment training to support the retention of flight and combat skills.

It is no easy feat to fly today's sophisticated and highly complex helicopters. Although there is a general assumption that the age of onboard computers has made the integration of flying and thinking easier, it has also made flying more demanding due to the interaction of those devices. The difficulty of interfacing with black boxes is further compounded by the environment in which crews must operate. For example, flying under the cover of darkness has become the preferred mode of operation, and as a result Night Vision Systems, both Night Vision Goggles (NVGs) and Forward Looking Infrared (FLIR), are commonplace.

Most aviators probably prefer to receive individual or crew training by flying the actual aircraft since this method affords obvious advantages over training devices. With today's technological advances, however, it is becoming exceedingly impractical to utilize actual aircraft to train certain critical skills. The use of actual aircraft is also constrained by safety and environmental concerns.

Therefore, how can a student or a rated aviator build, maintain, and challenge the correlated and interactive skills required to fight and survive on the battlefield of the future? Realistic, properly applied, and challenging training is the only answer. The overwhelming obstacle in achieving this goal is the sophistication of today's air and ground devices. An added complication is that many of todays systems require a tremendous amount of wide open space for realistic training. The military has been facing this growing problem for years, and has become increasingly reliant on a wide variety of simulation training devices as a solution.

To support the development of aviator skills, the Army employs a progressive training program which seeks to utilize the safest and most cost-effective tools available to facilitate each stage of training. This program uses classroom instruction, simulation training devices and actual aircraft. Figure 1 illustrates the generic hierarchy of training devices employed. The following section expands on the specific simulation devices used to support crew training for the various types of Army rotorcraft.

Simulation in the Army Multi-track Training Program

At Fort Rucker, Alabama, home of U.S. Army aviation, there is a wide diversity in the types of training students receive, depending on the type of helicopter. Throughout the training program, army aircraft simulation plays a vital role in the overall trained quality of the final product.

Within each individual aircraft qualification course, there is a hierarchy of required skills; individual training aids and simulation play a part. Some courses require more computer-assisted training than others. This usually depends on the complexity of the aircraft. Training requirements for most of today's advanced aircraft are complemented by a full fidelity simulator.

The first aircraft in which students are expected to "defeat the laws of gravity" is the UH-1 Huey. New recruits spend approximately 15 hours learning the basic skills of hovering, traffic patterns, normal flight maneuvers, some emergency procedures, and other basic flight tasks. Before this phase of training, however, the 2C35 procedural trainer is used to practice normal run-up procedures to include some related emergencies. Approximately 7.5 hours are logged, insuring that the individuals are proficient in the assigned tasks. The next simulator time they log is during the instrument phase. At this time they remain in the Huey for their actual flight portion, but the greatest amount of instruction takes place in the 2B24 Synthetic Flight Training System (SFTS). The 2B24 teaches both the basic and advanced instrument flying techniques. The actual aircraft is then used to demonstrate mastering the previously acquired skills. A total of 39 hours are logged in the 2B24 to learn and fine tune these new and important skills, while covering 28 of the 41 critical tasks. A critical task is defined as a skill that is required to be trained and maintained to a certain skill level. After the instrument phase, the new aviators are assigned an aviation track for Utility, Scout, or Attack vehicle. Figure 2 illustrates the various rotorcraft associated with Army's multi-track training program. Independent of the assigned track, simulation training devices play a vital role in the total training program. To outline this role, the following paragraphs will address each aircraft and the devices used to support the training requirements for that aircraft.

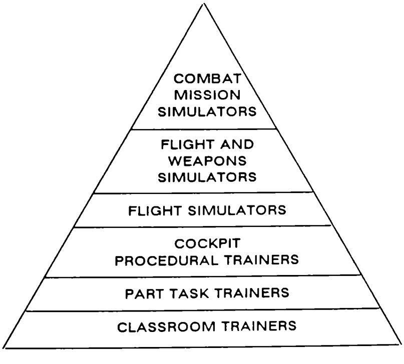


Figure 1 Army Aviation Training Device Hierarchy

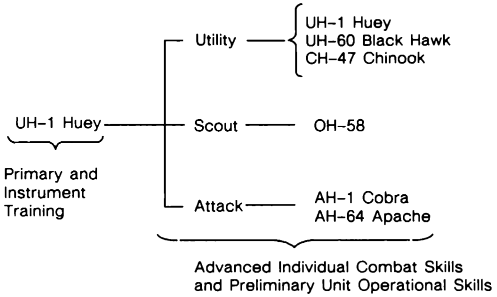

UH-1 Huey
Primary and Instrument Training
Utility
UH-1 Huey
UH-60 Black Hawk
CH-47 Chinook
Scout
OH-58
Attack
AH-1 Cobra
AH-64 Apache
Advanced Individual Combat Skills and Preliminary Unit Operational Skills

Figure 2 Army Rotorcraft Multi-Track Training

As stated earlier, the UH-1 uses a procedural trainer for initial entry skills along with an SFTS to teach the very demanding basic and advanced instrument proficiency. If tracked Utility, the individuals will proceed to one of three aircraft; these may be the UH-1, the more advanced UH-60 Black Hawk, or the heavy lift CH-47 Chinook.

If the student is tracked UH-1, he will return to the UH-1 SFTS for cockpit safety orientation during the Night Vision Goggle (NVG) phase.

The UH-60 and the CH-47 both have a full fidelity Flight Simulator (FS). These FS devices are capable of training 76 of the 93 critical tasks required of the UH-60 aviator, and 76 of the 106 critical tasks required of the CH-47 aviator. This training is accomplished in various modes, including: operating under different environmental conditions against hostile threats; performing shipboard operations; and performing NVG operations.

Most recently, the OH-58 model helicopter has entered the Army inventory as a scout vehicle. Although the Army does not possess a full scale simulator, it relies on both a Classroom Systems Trainer (CST) and a Cockpit Procedural Trainer (CPT) to supplement the training program. Individuals receive up to 108 hours on the CST during the transition phase. The device is a computer-based system with an interactive video disk used to teach both basic and critical OH-58 skills. After each four-hour block of instruction, the individual pilots utilize the CPT for a two-hour block of instruction to reinforce previously learned tasks. The CPT is not a flight simulator, but it does teach all systems operations along with emergency procedures.

The AH-1 Cobra helicopter is an attack tandem seat helicopter that carries a variety of onboard munitions. The AH-1 Flight and Weapons Simulator (FWS) was fielded in 1979. This full fidelity device consists of two components: a pilot station and a gunner's station, which can be operated independently or electrically connected to accomplish interactive crew training. It provides these capabilities and at the same time allows additional combined skills to be mastered. One example is the combined training of NVG flight and weapons deployment. Another is the use of Aircraft Survivability Equipment (ASE) and appropriate evasion techniques against a thinking, interactive intelligent threat. These threat models can be placed throughout the 80 x 100 km database to generate realistic and challenging scenarios.

Finally, the AH-64 Apache completes the family of U.S. Army helicopters. The AH-64 is the most sophisticated and complex helicopter in the world. To achieve a properly trained Apache crew member in the time allotted for the transition course, three trainers are used. First, students utilize the Cockpit, Weapons and Emergency Procedures Trainer (CWEPT), learning the basics of run-up, weapons initialization and firing, and procedures for some emergencies encountered in the real world. They receive approximately 15 hours throughout different phases of the transition course.

The next device used is the Target Acquisition and Designation Systems (TADS) Selected Task Trainer (TSTT). Due to the complexity of the AH-64, the TSTT is utilized to teach the numerous interactions the co-pilot gunner (CPG) must perform to employ the machine including operation of the fire control panel, weapons initializations, etc., as well as introducing the student to the doppler navigation system. The TSTT is only a CPG trainer, but it will be utilized throughout the CPG's career to help sustain the perishable skills required in the front seat. During the gunnery phase, all students receive a minimum of 7.5 hours training in this device.

The AH-64 Combat Mission Simulator (CMS) is one of the most sophisticated training devices in the field today. It consists of two elements: a pilot station; and a co-pilot gunner (CPG) station. Like the AH-1FWS, it allows training to be conducted independently or in an integrated crew configuration. Labeled as a combat mission trainer, it must replicate all tasks associated with the Combat Skills portion of the transition course. The CMS performs these functions and more. It challenges the crews to launch exacting munitions that are ballistically precise and kinematically detailed. It provides exacting scoring of all munitions fired by both friend and foe. While flying the CMS during day or night, in clear or adverse weather conditions, the crew can use all onboard sensors including Forward Looking Infrared (FLIR), Day TV (DTV), and Direct View Optics (DVO). All sensors are affected by the environmental factors that would normally be encountered, i.e., haze, fog, smoke, dust, and appropriate occulting factors. All onboard ASE equipment is simulated to an exacting degree of accuracy. This is one of the reasons this device is classified as SECRET. This level of fidelity, along with the accurate recording of the interface between the CMS and the interactive threat (up to 20 selected from a library of 77 models both friend and foe on the 80 x 100 km database), has proven to the remainder of the Army training community the value of a properly analyzed training program to be fielded with all new programs.

During the AH-64 transition, the CMS is heavily depended upon to teach those tasks that for a variety of reasons (safety, economics, real estate, etc.) cannot be taught in the actual aircraft. Two full fidelity CMSs are currently fielded at Fort Rucker, Alabama. During the transition phase, each pilot receives 13.5 hours learning basic and advanced gunnery skills, and is introduced to a small sampling of tactical employment of the Apache. During the post-graduate phase, the student receives an additional 15 hours of combat skills training. This covers the full spectrum of tactical employment and combined crew interactions.

The Army also has deployed flight simulators, flight and weapons simulators and combat mission simulators to support continuation training at field sites around the world.

As we have seen, the Army aviation community has learned to depend on simulation and training aids. These devices meet the Army's growing requirements for meaningful training in a cost-efficient manner. Figure 1 illustrated the hierarchy that has been structured by the utilization of these devices, from the lower order classroom trainers to the highest order CMS. Used in a building block approach, each succeeding level carries with it previously learned skills that will be applied at a more sophisticated level.

Future Advancements in Aviation Training

The proven success and benefits derived from well-structured and analyzed acquisition programs are demonstrated in the Army's reliance on training devices. The highly sophisticated RAH-66 Comanche program has proven this. Simulation has played a critical role in the design of the aircraft as well as the overall training requirements. Also, lessons learned from Desert Storm validate some of the earlier requirements placed on the RAH-66 training suite, mainly mobility. This requirement will be placed on the devices that will be delivered to field sites, while the Fort Rucker simulator will be a CMS with motion. Mobility will help insure that deployed units will be able to continue to train for their future combat missions, allowing them to retain very volatile and perishable skills.

The additional requirement for Air-to-Air (ATA) training posed a further challenge. The Comanche training solution was to incorporate a Helmet-Mounted Display (HMD) that projects the computer-generated database only inches away from the crews' eyes. This answers the concerns about transportability and allows the crew to have a database virtually 360 degrees around their aircraft while having a very large, instantaneous field of view.

The Comanche program will also involve the deployment of an in-cockpit embedded training system. This application of simulation directly in the aircraft will further enhance aviator sustainment training even to the point of providing last minute refresher training on the way to a battle.

In the future, the Army plans to exploit the benefits of the emerging Distributed Interactive Simulation (DIS) technologies through the Aviation Combined Arms Tactical Trainer (AVCATT) program. This program is evolving from the need for realistic combined arms training, that is, the need to expose and condition both experienced and inexperienced troops cooperatively to make rapid and accurate decisions in a hostile environment. They must be able to make precise judgement calls in highly stressful and demanding tactical situations. Eventually, as the Army implements its concept for a DIS Electronic Battlefield, aviator crews will be able to train in a large scale combat environment of hundreds to thousands of manned and automated participants. This environment will include simulators, simulations (such as wargames) and actual combat systems.

General George Patton stated: "There is no one solution for any tactical situation." War is a continually changing and unpredictable environment, and combatants must be able to adapt and change to meet the current flow of battle. Outside of actual battlefield conditions, the only method to hone these skills is to train in a simulated environment that realistically provides the stress and task loading that would be encountered during actual combat engagements.

About the Authors

Samuel N. Knight is a Senior Staff Scientist with CAE-Link Corporation in Binghamton, New York. Mr. Knight is the corporation's chief scientist for Distributed Interactive Simulation Initiatives and is the Principle Investigator for the Multiple Simulator Networking (MULTISIM) IR&D program. He is also a technical advisor on numerous projects including the Comanche program. He was previously the Tactical Systems Lead Engineer throughout the conceptual and development phases of the prototype AH-64 Combat Mission Simulator. During Desert Storm he was the Program Engineer for the AH-64 Desert STAARS program. He has over 22 years experience in visual and tactical systems simulation. He has a Master of Electrical Engineering degree from Cornell University and a B.S. in Electrical Engineering from Clarkson University. Mr. Knight has published work in the areas of tactical simulation, simulator networking, simulator fidelity, and mission rehearsal.

Bob Monette is a manager of International Marketing with CAE-Link Corporation of Binghamton, New York. Mr. Monette served in the U.S. Army from 1968 to 1991 and retired as a Chief Warrant Officer Four, Master Aviator. Since his arrival at CAE-Link in October of 1991, Mr. Monette has continued to pursue the international market in air, ground, and computer based training. He holds a degree in Aviation Science from Embry Riddle Aeronautical University. Mr. Monette has published work in the areas of simulator networking, simulator fidelity, and mission rehearsal.

AIAA-92-4133-CP

DOES A MOTION BASE PREVENT SIMULATOR SICKNESS?

Thomas J. Sharkey and Michael E. McCauley*
Monterey Technologies, Inc.
Carmel, California

Abstract

Simulator sickness is an unwanted side effect of many current flight simulators. Various strategies for minimizing the problem have been discussed by the NASA Steering Committee on Simulator Induced Sickness. One common proposal is to provide high fidelity motion cues to reduce the discrepancy or conflict between the visually-implied motion and the actual motion. The assumption is that the reduction in cue conflict will reduce the incidence and severity of simulator sickness. This hypothesis was tested using the NASA Vertical Motion Simulator (VMS) at Moffett Field. Ten pilots flew a UH-60 Blackhawk model in two low altitude flight maneuvers: S-turns and sawtooths. These flight tasks were selected because they had generated a very high incidence of simulator sickness in a previous study using a fixed-base wide FOV simulator. The pilots flew the maneuvers up to 60 minutes on each of two separate days, once with the motion base turned on, and once with the motion base off. Several types of data were collected including periodic self report, pre-post symptom checklists, dark focus, and pre-post postural equilibrium tests. The results indicate that the motion base condition did not result in a lower incidence or severity of simulator induced sickness.

Introduction

Simulator induced sickness is a problem in many military flight trainers and research simulators.[1,2] Simulator sickness is the appearance of motion sickness-like symptoms as a result of a simulated flight when the symptoms would not occur as a result of performing the same maneuvers in the actual aircraft.

One of the most appealing potential explanations of simulator sickness is the conflict hypothesis.[3] The conflict hypothesis is not specific enough to derive quantitative predictions, but may be interpreted as predicting that the incidence of simulator sickness is a function of the amount of cue conflict. A reduction in the magnitude of the conflict between the motion indicated visually and the motion sensed by the vestibular, proprioceptive, and other sensory systems should, according to this logic, reduce simulator sickness.

Presumably, a motion base reduces this conflict by providing the pilot rotational and translational acceleration cues which are consistent (in direction, if not in magnitude) with the visual acceleration cues. A fixed base simulator lacks the capability to provide these motion cues, and therefore, lacks the capability to reduce the cue conflict.

One characteristic of many motion bases that would tend to limit their ability to reduce the sensory conflict is motion washout. Motion washout is the acceleration applied to the cab to keep it from approaching the translational or rotational limits of the system. Since it is not uncommon attenuate the motion of the cab prior to the cessation of the visually described accelerations, false cues can be introduced. That is, the cab actually may be accelerated in the opposite direction from the simulated aircraft. Typically, accelerations also are applied to the cab to return it to the center of its range of motion. These anomalous accelerations are intended to be below the pilot's threshold, and, thus, it is hoped that they do not to contribute to sensory conflict.

This motion/no motion (MONOMO) experiment tests the hypothesis that a large scale motion base will reduce simulator sickness by reducing inter sensory conflict. The pilots flew the same tasks in two motion conditions: fixed base and with a large 6 degree of freedom motion base. It was predicted that the motion cues would reduce the inter sensory cue conflict relative to the fixed base condition. This reduced conflict was expected to result in a lower incidence and/or severity of simulator induced sickness in the motion condition than in the fixed base condition.

----

* Member AIAA.

Method

Pilots. Ten helicopter pilots participated in this experiment. All of the men were volunteers. Nine were U.S. Army pilots based at Ft. Ord, California. The tenth was a helicopter pilot recently separated from the Army. They were treated in accordance with existing NASA and Army guidelines for the use of human subjects. Four of the pilots were currently flying UH-60s. The other pilots were flying AH-1s, UH-1s or OH-58s. The aviation backgrounds of the pilots are outlined in Table 1.

Procedure. Each pilot flew the Vertical Motion Simulator (VMS) twice; once in fixed base mode and once in VMS Nominal mode. The flights were more than a week apart for seven of the pilots. Due to scheduling constraints, the flights made by two of the pilots were on consecutive days. One day separated the flights of the remaining pilot.

TABLE 1. AVIATION BACKGROUND OF THE PILOTS PARTICIPATING IN THE MONOMO EXPERIMENT.

	MEAN	σ	MAX	MIN
AGE (YEARS)	26.3	2.6	31	23
HOURS IN TYPE	384.5	334.6	1000	55
TOTAL ROTARY WING HOURS	534.6	395.9	1400	220
TOTAL FIXED WING HOURS	66.8	188.0	600	0
TOTAL FLIGHT HOURS	601.4	556.5	2000	220
TOTAL SIMULATOR HOURS	80.9	32.1	140	45
VISUAL SIMULATOR HOURS	31.1	33.4	100	0
NON-VISUAL SIM. HOURS	49.8	21.2	88	20

Prior to their first flight, each pilot was briefed on the purpose of the experiment, the tasks to be performed, and the procedure to be followed. Each pilot read a manual describing the VMS safety procedures, and received a thorough briefing on safety procedures. Each pilot completed a set of questionnaires regarding their aviation background, current state of health, and history of motion and simulator sickness.

Pilots were individually fitted with Integrated Helmet and Display Sighting System (IHADSS) helmets. The IHADSS helmet was used to measure the pilot's head movements. The monocle used to display PNVS imagery and symbology in conjunction with the IHADSS helmet was not used other than to allow the pilot to align the helmet (i.e., to perform a boresight alignment procedure) prior to flight. The monocle was not mounted on the helmet during the simulated flights.

Each pilot completed a battery of tests of postural equilibrium before, immediately after, and again 30 minutes after the conclusion of each flight. This battery consisted of three replications of both the stand-on-leg eyes closed (SOLEC) test and the walk-on-floor eyes closed (WOFEC) test.[4] The maximum values for the SOLEC and WOFEC in this experiment were 30 seconds and 12 steps, respectively. The mean of the three replications of the SOLEC and WOFEC was used in the statistical analyses.

After completing each of the tests in the postural equilibrium battery the state of each pilot's accommodation was measured using a stigmatascope.[5] The pilot adjusted the stigmatascope until the small light appeared to be in focus. This adjustment was done three times before, after, and again 30 minutes after each flight. The mean of the three adjustments of the stigmatascope at each administration was used in the statistical analyses. Pilots also completed a Simulator Side Effects Questionnaire (SSEQ) before, after, and 30 minutes after each flight.[6] The SSEQ allowed pilots to indicate the symptoms, if any, that they were experiencing at the time the questionnaire was completed.

During each flight, the pilot verbally reported his well-being on a 7-point scale in response to visual and auditory prompts. These prompts occurred every minute during the flight. A report of "1" indicated that the pilot felt normal and was not experiencing any unusual or adverse symptoms. A "7" indicated that the pilot was unable to continue due to discomfort. Intermediate values were used to report well-being between the endpoints. Pilots who vomited or were otherwise unable to continue were assigned a "7". When a pilot reported or was assigned a "7" the flight was ended.

Tasks. The pilots performed two flight tasks during the experiment: S-turns and a sawtooth. (The later task is sometimes described as a lateral mask-unmask maneuver.) The tasks were chosen for this study had made 100% (6 of 6 pilots) simulator sick in a previous experiment with a fixed base flight simulator, and because the maneuvers placed a relatively low demand on a motion base.[7]

These tasks were performed in the vicinity of a simulated taxiway. The taxiway and the traffic cones on it are described below. Pilots were instructed to fly both of the maneuvers at approximately 15 ft ± 5 ft AGL. The maneuvers were self paced.

Pilots were advised that others who had already flown the slalom reported it to be difficult to perform at airspeeds greater than 20 to 25 kts maximum. They were also advised that pilots not familiar with the UH-60 had a strong tendency to over-control the aircraft.

Figure 1 shows idealized ground tracks of the sawtooth and s-turn maneuvers. The small rectangles indicate the positions of the traffic cones which were placed on the taxiway.

During the s-turn maneuver, the pilot flew over the first pair of cones, then made a 180° turn to the left and flew back over the taxiway between pairs of cones. After passing between pairs of cones a 180° turn to the right was made and the aircraft flown over the next pair of cones. The pilots repeated this process until reaching the end of the taxiway.

For the sawtooth (or lateral mask-unmask) maneuver the pilot initially hovered the aircraft over the cone on the left side of the taxiway at 15 ft AGL with the heading of the aircraft coinciding with the taxiway. The pilot then side-slipped the aircraft forward and to the right, while maintaining a constant heading, until it was over the right hand cone in the next pair. The aircraft was then stopped and side-slipped directly across the taxiway to the left, again while maintaining a constant heading. Once over the left-hand cone, the aircraft was stopped, and maneuvered forward and to the right. This procedure was repeated until the end of the taxiway was reached. At the end of the taxiway the pilot positioned the aircraft for the next maneuver as directed by the experimenter.

Pilots alternated between these two tasks for approximately 60 minutes, or until they reported or were assigned a "7".

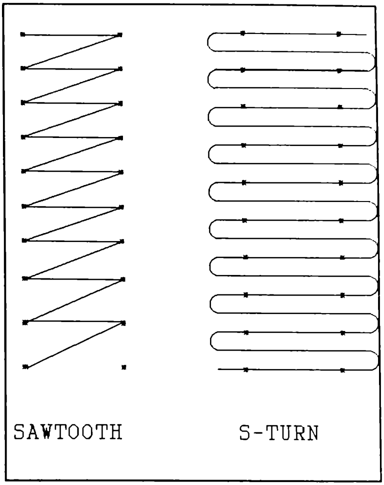


Figure 1. Idealized ground track of S-turn and Sawtooth maneuvers.

Motion Base. NASA's Vertical Motion Simulator (VMS) motion base was used in this experiment to provide motion cues to the pilots.[8] The VMS is located at the Ames Research Center, Moffett Field, CA. For this experiment, the VMS motion system parameters were adjusted to maximize the magnitude of both translational and rotational motion cues while limiting the frequency of encountering the software or hardware limits of the motion system. With the exception of the yaw gain, which was reduced in the s-turn maneuvers to accommodate the 180° turns, the motion base parameter settings were the same during these tasks.

Table 2 describes the software motion, velocity, and acceleration limits of the VMS. Table 3 describes the frequency response.

Visual System. The out-the-window visual scene was created with a Singer-Link DIG image generation system. Due to the density of edges in the scene, the image normally updated at a 30 Hz rate. However, on the occasions when there were few edges visible the system would automatically shift to a 60 Hz update rate.

TABLE 2. NASA VMS PERFORMANCE ENVELOPE SOFTWARE LIMITS.

Axis	Position (ft or rad)	Velocity (ft/sec or rad/sec)	Acceleration (ft/sec/sec or rad/sec/sec)
Longitudinal	3.0	4.0	10.0
Lateral	15.0	8.0	13.0
Vertical	22.0	15.0	22.0
Roll	0.24	0.7	2.0
Pitch	0.24	0.7	2.0
Yaw	0.34	0.8	2.0

TABLE 3. NASA VMS FREQUENCY RESPONSE.

Axis	Frequency	
	With Feedforward	Without feedforward
Longitudinal	No feedforward used	0.8 Hz
Lateral	1.8 Hz	0.1 Hz
Vertical	1.2 Hz	0.2 Hz
Roll	2.1 Hz	0.2 Hz
Pitch	1.9 Hz	0.8 Hz
Yaw	3.0 Hz	0.9 Hz

There were four separate out-the-window displays in the cockpit. Three of the displays were arranged in front of the pilot at approximately eye height. The fourth display was a chin window. The chin window was located in the lower right portion of the cab. The horizontal field of view was approximately 150°. The vertical field of view, exclusive of the chin window, was approximately 27° (12° up and 15° down from the design eye position).

Gaming Area. The gaming area consisted of a north-south runway and a parallel taxiway. There were a number of large buildings in the immediate area as well as some buildings in the far distance. A variety of small objects (e.g., cars, trucks, trees, people) at random locations on either side of the taxiway. The gaming area did not have any topographical features such as mountains or hills. That is, the area was flat.

The maneuvers were flown in the vicinity of the taxiway. There were 10 pairs of 3 ft tall traffic cones on the taxiway. Pairs of cones were spaced every 120 ft along the length of the taxiway. The distance across the taxiway between cones in a pair was 80 ft. This spacing is similar to the spacing of taxiway lights on a portion of Moffett Field used in an earlier study.[7] The surface of the taxiway resembled a series of concrete or tarmac slabs separated by expansion joints. Tire skid marks were placed on the surface of the taxiway to provide additional visual cues.

Cockpit Instrumentation. All of the flight indicators available to the pilot were presented on a Heads-Up Display (HUD). Along the top of the HUD was a heading tape and lubber carrot. Altitude AGL was shown on the right side of the HUD. Inertial velocity ("VEQ") was shown along the left. Both analog tapes and digital readouts were presented for altitude and velocity. A Vertical Speed Indicator (VSI) was positioned to the left of the altitude tape. The "W" symbol in the upper center of the HUD represented ownship. The pitch ladder was presented in 5 degree intervals. No other flight or systems information was available to the pilots.

A question mark ("?") appeared in the center of the HUD at 1 minute intervals. It was accompanied by a tone that lasted approximately 3/4 second. The question mark and tone prompted the pilot to report his well-being on the 7-point scale. The question mark symbol was removed from the HUD by the experimenter after the report was recorded.

Cockpit Controls. The cyclic and collective were UH-60 equipment. The rudder pedals and toe brakes were generic equipment. All of the controls were hydraulically operated. The only instrument on the cockpit panel was a collective position indicator. The panel was otherwise blank.

Dependent Measures

Subjective and objective measures of simulator sickness were obtained. The subjective measures included pilot reports of well-being during the simulated flight, and reports of symptoms pre, immediately post flight, and 30 minutes post flight.

The objective measures of simulator sickness were the amount of time the pilot could stand on one leg with his eyes closed, the number of heel-to-toe steps he could take along a line, and a measure of dark focus. The objective measures were recorded pre flight, immediately post flight, and 30 minutes post flight.

Results

Self Reports of Well-Being. Simulator induced sickness was observed during this experiment. In the fixed base condition, three of the 10 flights were terminated due to the pilot scoring a "7". In the VMS nominal motion condition four of the 10 flights in the VMS nominal motion condition were terminated for this reason. On only one flight out of the 20 total were absolutely no symptoms reported. The average maximum value of self reported well-being in the fixed base and VMS nominal motion conditions were 3.8 and 4.5, respectively. The direction of this difference is not consistent with the conflict hypothesis. An Analysis of Variance (ANOVA) performed on these data indicates that the difference is not statistically significant ($F_{1,9} < 1.0$).

The average time from the beginning of each flight to the first increase in the pilots self report of well-being was 23.7 minutes in the fixed base condition, and 21.8 minutes in the VMS nominal condition. This difference is not statistically significant ($F_{1,9} < 1.0$).

The average time to the maximum self report of well-being was 30.9 minutes in the fixed base condition and 35.3 in the VMS nominal condition. This difference is not statistically significant ($F_{1,9} < 1.0$).

The time histories of the self reports of well-being from each pilot in each motion condition were inspected visually. No systematic differences attributable to the motion condition manipulation were observed.

Ataxia.

WOFEC- Figure 2 shows the average number of steps taken by the pilots before, immediately post, and 30 minutes post flight in each motion condition. An ANOVA indicates that there is a significant interaction between motion condition and time-of-test ($F_{2,45} = 3.845$, $p \leq 0.03$). Post-hoc tests indicate that the pilots took fewer steps immediately post-flight in the fixed base condition (5.8 steps) than in the VMS nominal condition (7.4 steps) ($F_{1,9} = 4.558$, $p \leq 0.06$), and that they took more steps during the 30 minute post flight test in the fixed base condition (6.9 steps) than in the VMS nominal condition (5.3 steps) ($F_{1,9} = 3.561$, $p \leq 0.09$). The preflight difference between motion conditions is not statistically significant.

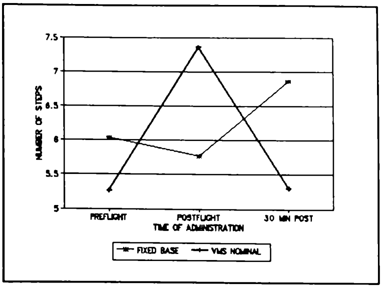


Figure 2. Mean pre-, post-, and 30 minute post flight WOFEC scores in fixed base and VMS nominal motion conditions.

SOLEC- Figure 3 shows the mean number of seconds that the pilots were able to perform the SOLEC test pre-, immediately post-, and 30 minutes post flight in each of the motion conditions. An ANOVA indicated that neither main effect, nor the interactions are statistically significant.

Stigmatascope. Figure 4 shows the mean dark focus of the pilots pre-, immediately post-, and 30 minutes post flight in each motion condition. An ANOVA revealed a significant difference between motion conditions ($F_{1,45} = 4.311$, $p \leq 0.05$). In the fixed base condition the mean is 1.65 diopters. The mean in the VMS nominal condition is 2.0 diopters. The effect of time of measurement and the interaction between motion condition and time of measurement are not statistically significant.

SSEQ. Figure 5 shows the average Total score of the SSEQ pre-, immediately post-, and 30 minutes post flight.[6] An ANOVA indicates that there was a main effect of time of test administration ($F_{2,18} = 8.629$, $p \leq 0.01$). The effect of motion condition, and the interaction between time of test administration and motion condition, were not statistically significant.

Post hoc tests indicate that the immediate post flight SSEQ scores are greater than the preflight scores in both the fixed base and VMS nominal motion conditions ($p \leq 0.01$ in both cases). In addition, the immediate postflight score and the 30 minute post flight score are significantly different in the VMS nominal motion condition ($p \leq 0.05$).

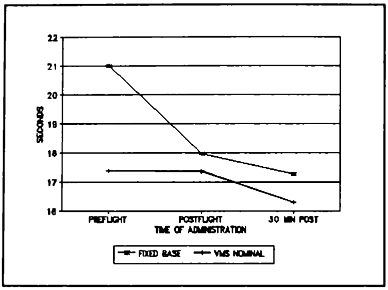


Figure 3. Mean pre-, post-, and 30 minute post flight SOLEC score in fixed base and VMS nominal motion conditions.

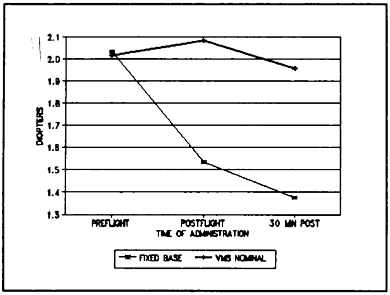


Figure 4. Mean number of diopters required pre-, post-, and 30 minutes post flight in fixed base and VMS nominal motion conditions.

Statistically significant effects of time of administration are also present in the Nausea subscale ($F_{2,18}$ = 8.108, $p \leq 0.01$); the Visual subscale ($F_{2,18}$ = 6.960, $p \leq 0.01$); and in the Disorientation subscale ($F_{2,18}$ = 5.651, $p \leq 0.02$). The effect of motion condition and the interaction between motion condition and time of administration are not significant on any of the subscales.

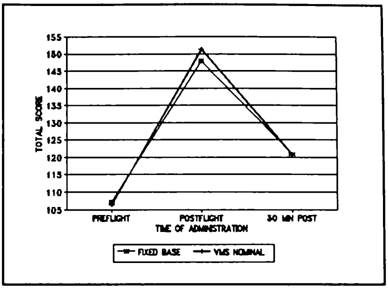


Figure 5. Mean SSEQ total score pre-, post-, and 30 minutes post flight in the fixed base and VMS nominal motion conditions.

Discussion

Contrary to our prediction, the motion base condition did not reduce simulator sickness. The expected differences in the incidence and severity of simulator induced sickness between motion conditions were not observed. Pilots showed evidence of sickness in both the motion and no-motion conditions.

Other possible reasons for the failure of this experiment to reveal an effect of the motion cue manipulation, if an effect existed, include:

(1) The measures were insufficiently sensitive to detect a reduction in sickness attributable to the reduction in sensory conflict.

A power analysis performed prior to the study indicated that there was better than an 80% probability of detecting a 1-point difference in self reports of well-being with this size sample. In addition, the SSEQ scores show good sensitivity to the onset of, and recovery from, simulator sickness (see fig. 5). Together these suggest that the sensitivity of the measures was adequate to detect meaningful differences.

(2) The difference in conflict between the fixed base and VMS nominal conditions in this study was not

large enough to effect the incidence or severity of simulator sickness.

Calculation of the conflict between the acceleration signalled by the visual system and the cues provided by the motion system has not been completed. We would note that the problem of quantifying the magnitude of the sensory conflict is not a simple one. It is possible to calculate the difference in acceleration signalled by the visual system and the acceleration provided by the motion base in each degree of freedom at each instant. (In the fixed base situation, the conflict would simply be the acceleration signalled by the visual system.) However, this type of metric does not consider the efficacy of the visual cues. For example, one would suspect that the pilot would tend to under-perceive the magnitude of the visually signalled accelerations when the scene is sparse relative to conditions where the visual scene is rich in texture. The inter-sensory conflict would effectively be less with the sparse visual scene than with the rich scene for the same accelerations.

Another aspect of the conflict that is problematic is the integration or combination of conflict measures from different degrees of freedom. As an example, consider a situation where the accelerations of the cab are in the same direction as the visually described accelerations in five degrees of freedom, but the acceleration in the other degree of freedom is either in conflict, or the cab motion is below threshold and the visual acceleration is above threshold. The question that comes to mind is: Is this a cue conflict situation?

In any event, there would seem to be little reason to predict that future flight trainers will have greater motion cuing capability than the VMS has. Consequently, if the current capability is insufficient to reduce conflict and simulator sickness, then other engineering and simulation management solutions to the problem must be developed.

(3) The false cues in the motion condition may have had an adverse impact roughly equivalent to absence of cues in fixed base condition. In other words, it may be that the "good" and "bad" motions effectively canceled each other out in the VMS nominal condition. In the fixed base condition, there would not have been any "bad" motion, and there wouldn't have been any "good" motion. Sickness in the fixed base condition may be due to the lack of vestibular confirmation or corroboration of the acceleration, rather than as a response to "bad" motion.

It should be noted that the tasks used in this experiment had such a low dominant frequency in the translational degrees of freedom (on the order of seconds in both tasks) that the washout and return to center, both of which are accelerations in conflict with the visually signalled accelerations, may have been large proportions of the total cab accelerations in the VMS nominal condition.

This study attempted to identify an engineering solution to the problem of simulator induced sickness. The results do not support the hypothesis that a large, capable motion-base will ameliorate the problem. Recent evidence indicates that engineering "solutions" may be limited to items such as calibration of optical systems, tuning flight dynamics and handling qualities, and avoiding large transport delays. Simulation management solutions include reducing exposure time for initial flights, increasing altitude (to reduce Global Visual Flow), educating flight surgeons and simulator instructors, and limiting post-exposure activities to reduce risk in driving automobiles or flying.

References

1. Kennedy, R.S., Dutton, B., Lilienthal, M.G., Ricard, G.L. & Frank, L.H. Simulator Sickness:Incidence of simulator aftereffects in 10 Navy flight trainers. Paper presented at the SAFE symposium, Las Vegas, NV. 1984.

2. Kennedy, R.S., Lilienthal, M.G., Berbaum, K.S., Baltzley, D.R., & McCauley, M.E. Simulator sickness in U.S. Navy flight simulators. Aviation, Space & Environmental Medicine, 60:10-16, 1989.

3. Reason, J.T., & Brand, J.J., Motion Sickness. London: Academic Press. 1975.

4. Thomley, K.E., Kennedy, R.S. & Bittner, A.C. Development of postural equilibrium tests for examining simulator aftereffects. Perceptual and Motor Skills, 63:555-564, 1986.

5. Borish, I.M. Clinical Refraction (3rd ed.). Chicago, Il: The Professional Press, Inc. 1975.

6. Lane, N.E. & Kennedy, R.S. A New Method for Quantifying Simulator Sickness: Development and Application of the Simulator Sickness Questionnaire (SSQ). Technical Report EOTR 88-7, Essex Corporation, Orlando, FL. 1988.

7. Hennessy, R.T., Sharkey, T.J., Matsumoto, J.A. & Voorhees, J.W. Simulator induced alteration of head movements (SIAHM). Paper presented at the AIAA Flight Simulation Technologies Conference. Hilton Head, SC. 24-26 August 1992.

8. Danek, G. Vertical motion simulator familiarization guide: components and systems. NASA Ames Research Center, Moffett Field, CA. 1991.

Acknowledgement

This work was supported by the U.S. Army Aeroflightdynamics Directorate, Contract Number NAS2-12927. Ms. K. Perlaki served as the Contracting Officer's Technical Representative.

The opinions expressed in this report are those of the authors and do not reflect the official views of the Army or of NASA; and they do not constitute a standard, specification, or regulation.

AIAA-92-4134-CP

SIMULATOR INDUCED ALTERATION OF HEAD MOVEMENTS (SIAHM)

Robert T. Hennessy
Thomas J. Sharkey
Joy A. Matsumoto

Monterey Technologies, Inc.
Carmel, CA.

and

James W. Voorhees

Advanced Simulation Engineering
Vancouver, WA

ABSTRACT

The purpose of this research project was to determine if exposing pilots to simulated helicopter flight known to induce simulator sickness causes alterations in the normal pattern of head movements exhibited by the pilots during actual flight. To test this hypothesis, head movements were measured while the pilot flew the NASA AH-1S Flying Laboratory for Integrated Test & Evaluation (FLITE) Cobra through maneuvers emphasizing turning and lateral movement of the aircraft, e.g., S-turns, and a sawtooth path involving lateral and forward progressive hovering movement. These maneuvers were performed on a taxiway at an altitude of twenty-five feet. Between AH-1S flights, the same maneuvers plus a short pursuit flight were executed in a fixed-base helicopter simulator, the U.S. Army's Crew Station Research and Development Facility (CSRDF). Participants were six, qualified, U.S. Army AH-1 pilots with no or little exposure to visual flight simulation.

All six pilots experienced moderate to severe simulator sickness while performing the maneuvers in the CSRDF. Comparison of head movement data taken in the AH-1S before and after the simulator flight shows marked reduction in yaw head movements during the post simulator flight.

INTRODUCTION

Simulator Induced Sickness (SIS) is a phenomena that results in symptoms such as nausea, headache, eyestrain, and general discomfort in pilots flying certain maneuvers in a simulator. SIS is different from classic motion sickness because a pilot performing similar maneuvers in an aircraft would not experience those symptoms. Although SIS seems to have the same symptoms as motion sickness it has two characteristics that separate it from that class of sickness. First it tends to occur more often, and more severely, in more experienced pilots. Second, in most cases, the symptoms of SIS will largely disappear over several flights in a particular simulator if the simulator flight periods are close together in time.[1]

While the latter characteristic might at first appear positive the phenomenology behind the "disappearance" of symptoms is of great concern to those who use simulators for training, research and development (R&D), or test and evaluation (T&E). The concern is that pilots, in an attempt to reduce the effects of the perceived SIS, are adopting behaviors that they would not use in the real aircraft being simulated - therefore learning inappropriate responses. These inappropriate responses, while positive in allowing the pilot to perform in the simulator, may reduce positive transfer of training and calls into question any results generated by the simulator to be used in R&D or T&E applications.

Some observational reports indicate that pilots begin to restrict head movement in simulators (and aircraft like the AH-64 which employ an artificial viewing device with its Forward Looking Infra Red (FLIR) sensor) when they begin to feel the nausea, headache, and other symptoms associated with SIS. This might be because of the proposed vestibular involvement in both motion sickness and SIS. If, in fact, reduced head movements is one of the inappropriate responses learned because of SIS, this could be a major problem with simulator training for engagements such as air-to-air combat.

In 1987 NASA, along with the U.S. Army, formed a committee to bring together the data that had been collected regarding SIS, and plan a series of research experiments that might systematically address the problems generated by SIS.[2] The committee defined as one of its goals finding answers to the following questions:

1) Are the pilots responses to selected maneuvers (particularly head movement) in the simulator really different than in the aircraft, and are they different after exposure to the simulator?

2) Are the maneuvers that cause SIS in the simulator, particularly those involving a lot of head movement, ever done in a real aircraft, and do these maneuvers cause any symptoms when the pilot performs them in the real aircraft?

It was felt that these questions were basic to determining a firm baseline to examine the problem of Simulator Induced Sickness.

The impetus for this study came from these questions and the fact that the equipment to do this type of study exists at the Army/NASA facility at Ames Research Center. The most reasonable approach to answering the above questions was to conduct a study where the pilots flew a set of maneuvers in an aircraft, then flew them in a simulator, then flew the set immediately in the aircraft again.

This study did not attempt to address all of the concerns surrounding SIS. Variables such as pilot age/experience, complexity of the simulator visual system, range of maneuvers, and time to extinction of the SIS symptoms in the simulator were not tested. The study was specifically constructed to answer the two questions shown above, and to form a baseline for a more complex program of simulator/aircraft SIS research.

METHOD

Subjects. Four Army helicopter pilots and two Air National Guard helicopter pilots, all men, served as the subjects for this study. All pilots were qualified to fly AH-1 Cobra helicopters.

Apparatus.

Crew Station Research and Development Facility. The Crew Station Research and Development Facility (CSRDF) located at NASA Ames Research Center is an advanced rotorcraft research simulator with a unique blend of components. The simulator consists of a two-seat cab on a fixed platform.[3,4] The pilot's visual imagery is produced by a fiber optic display mounted on his helmet. Wide-angle eyepieces fit closely over the pilot's eyes producing a large, high-resolution image. With his head in one position, the pilot has a field-of-view measuring 40 degrees vertical by 120 degrees horizontal. (See Figure 1.) Motion of the pilot's head is tracked by an infrared device in the helmet, allowing the computer to display the correct image wherever he looks. The resulting field-of-regard is unlimited.[3]

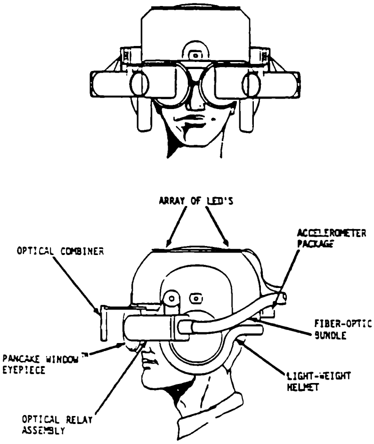


Figure 1. Wide Field-of-View Fiber Optic Helmet Mounted Display.

AH-1S FLITE Helicopter. The AH-1S FLITE helicopter is a two-place tandem seat and single-engine platform. The aircraft has been modified with the addition of the Pilot Night Vision System (PNVS), on-board computers, and a laser tracking reflector. For a more detailed description of the AH-1S FLITE helicopter, see Haworth, et al.[5]

Procedure.

Pilots flew two flights in the AH-1S FLITE helicopter and two flights in the Crew Station Research and Development Facility (CSRDF) advanced helicopter simulator. Pilots were asked to perform tests of postural equilibrium and eye focus and to complete a questionnaire on simulation side effects before and after each helicopter and simulator flight. A discussion of these results is beyond the scope of this paper.

AH-1S FLITE Helicopter Pre-Simulator Flights. The pilots were required to fly one familiarization flight with a Safety Pilot in the Army/NASA AH-1S FLITE helicopter prior to data collection. When the safety pilot was assured that the subject pilot was proficient with this particular helicopter, the data collection phase of the study began. Each pilot was briefed on the maneuver requirements and trained to fly the maneuvers before data collection began.

Pilots were required to perform two repetitions each of two different maneuvers along a taxiway. Edge lights on the taxiway were used to guide the pilots in their performance of the maneuvers. The edge lights were 80 feet apart. Each pair of lights was 180 feet away from the next closest pair. Ten pairs of lights defined the maneuver space.

The first maneuver was called the Sawtooth maneuver and the second maneuver was called the S-Turn maneuver. The maneuver patterns are shown in Figure 2. Both Maneuvers were performed at 25 feet above ground level (AGL). The Sawtooth maneuver required that the pilot fly diagonally forward from the first light on the left side of the taxiway, while maintaining heading parallel to this taxiway. Then, the pilot flew laterally from the second light on the right side of the taxiway directly to the second light on the left side. This pattern was repeated until the pilot was at the last cone on the left side of the taxiway. Each pilot performed the Sawtooth maneuver twice before proceeding to the next maneuver. Pilots were instructed to maintain a reasonable speed that would allow them to perform the maneuver in a precise manner, but not to come to a complete stop. Each repetition of the Sawtooth maneuver took about 3.5 minutes.

The next maneuver the pilots performed was the S-Turn. The pilot again positioned the helicopter at the first light on the left side of the taxiway, heading perpendicular to the taxiway, i.e., pointed at the first light on the right side. Pilots flew across the taxiway over both sets of lights, before turning 180 degrees to the left. After the turn, the pilots flew across the taxiway passing between the first and the second sets of lights, turned 180 degrees to the right and flew over the second set of lights. This pattern was repeated until the pilot flew across the last set of lights (from left to right) on the taxiway. Each pilot performed the S-Turn maneuver twice. Pilots were instructed to maintain a reasonable speed that would allow them to make the 180 degree turns with a moderate amount of aggression. Each repetition of the S-Turn maneuver took about 4½ minutes.

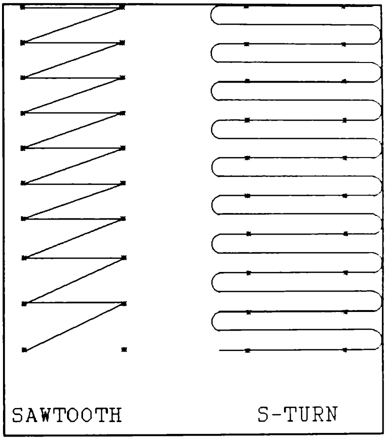


Figure 2. Idealized Ground Track of Sawtooth and S-turn Maneuvers.

To avoid flying the AH-1S Cobra in a tailwind, pilots performed each helicopter maneuver in the same direction. Following the completion of each maneuver, the pilot flew back down the taxiway and assumed the same starting position for the next maneuver. The safety pilot flying with the subject pilot rated the pilot's performance at the end of each maneuver. The safety pilot ratings indicate that the pilots all performed the maneuvers as instructed. An onboard computer recorded pilot head position (in azimuth and elevation) and aircraft heading, ground speed, altitude MSL, roll, and pitch. Data were collected at approximately 15 Hz.

CSRDF Simulator Flights. The subject pilots were introduced to the CSRDF advanced helicopter simulator. During a 45-minute familiarization and practice session, each pilot received flight instruction. If a pilot wished, he was allowed to practice the two maneuvers on a runway in the simulator. The simulator runway was used as the aircraft taxiway for the simulator flights. Instead of lights, traffic cones were portrayed along the runway with the same spacing as the lights on the taxiway for the helicopter flights; the cones were 80 feet across from one another, and 180 feet separated each pair of cones. Ten pairs of traffic cones defined the maneuver space in the simulator.

On the day after the simulator familiarization and practice, the simulator data collection phase of the study began. Each pilot was required to repeat the same maneuvers in the CSRDF simulator as he performed in the AH-1S FLITE helicopter. Two repetitions of the Sawtooth maneuver were followed by two repetitions of the S-Turn maneuver. Following completion of each maneuver, the pilot was prompted to report the degree of discomfort, i.e., SIS, he was feeling using a seven-point well-being scale. Between maneuvers, the pilot flew back down the runway to the starting position for the next maneuver.

After all four maneuvers were completed, the pilot flew a 12 minute pursuit task flying at an altitude of 100 feet AGL and at an airspeed of 120 knots. The pilot was required to follow a Hind helicopter at a distance of between 250 and 350 feet. The slant range distance to the Hind helicopter was displayed digitally on the Helmet Mounted Display (HMD). When the pilot lagged more than 350 feet behind the Hind, a green triangle pointing up appeared above the range display. When the pilot closed to less than 250 feet behind the Hind, a red triangle pointing down appeared below the range display. The pilots were instructed to follow the ground path that the Hind traversed and not cut corners. At two-minute intervals throughout the pursuit task, each pilot was prompted to report how he was feeling using a seven-point well-being scale to be described presently.

At the end of the pursuit task, each pilot again was scheduled to fly two repetitions of the two lateral Sawtooth and S-Turn maneuvers that they had flown in the AH-1S Cobra helicopter. Immediately following the last maneuver in the simulator, the pilots performed the postural and vision tests and filled out the simulation side effects questionnaire.

AH-1S FLITE Helicopter Post-Simulator Flight. As soon as the tests were completed, each pilot entered the AH-1S Cobra helicopter and flew two repetitions of the two maneuvers. As before, the pilots were rated on their performance by the Safety Pilot.

Well-Being Scale. The Well-Being Scale was presented to the pilots during simulator flights only. Figure 3 illustrates the well-being scale as it appeared on one of the screens in the simulator cockpit. The well-being scale consists of seven buttons with the numbers 1 through 7; it was always displayed on the screen. When the pilot was prompted for a response, he touched the button which corresponded to his sense of well-being at that time. One corresponded to: "I feel fine and completely symptom-free" and 7 corresponded to: "I feel as though I am too uncomfortable to continue." The numbers 2 through 6 were used to indicate intermediate levels of well-being.

1	2	3	4	5	6	7

Figure 3. Schematic of Well-being report boxes. These were displayed to the pilot on a touch screen in the cockpit throughout the flight.

Pilots were instructed that they need not wait for the prompt to report any sensation of unpleasant symptoms. Pilots were asked to provide a well-being report before and after each maneuver in the simulator. A red question mark appeared in the middle of the Helmet Mounted Display when a response was required. The question mark remained on the display until the pilot pressed a button representing his state at that time. During the pursuit flight, pilots were asked to provide a well-being report before and after they started the pursuit flight, and at two-minute intervals throughout the pursuit flight. The red question mark was displayed on the Helmet Mounted Display to prompt the pilots to make the report. Pilots were instructed that they could stop the simulation at any time they started to feel ill, and did not wish to continue.

RESULTS

Self-ratings by the pilots with the well-being scale are the primary data on the degree of SIS experienced. The performance data of direct interest are the extent of head movement in the azimuth or yawing plane in the AH-1S Cobra before and after the simulated flight in the CSRDF. A second measure of interest is the time to perform the Sawtooth and S-turn maneuvers. Data from the simulator and the performance tests are outside the scope of this report and will not be discussed further.

SIS and Well-being ratings. Table 1 presents the well-being rating reported after completion of each maneuver in the simulator. Some data are missing because none of the six pilots was able to complete the scheduled series of maneuvers in the CSRDF. Five of the six pilots asked to quit the simulation because they were feeling too ill to continue. The sixth pilot was unable to complete the scheduled series of maneuvers because the simulator failed. However, this pilot remarked that he would probably not have been able to finish the simulation session. The last number shown for each pilot indicates where in the series the simulation was terminated.

Table 1. Well-being Ratings at the End of Each Flight Maneuver in the Simulator.

	TASK								
Pilot	Saw1	Saw2	Stn1	Stn2	Chase	Saw3	Saw4	Stn3	Stn4
P1	2	3	4	4	6	6	6	(x)	
P2	1	1	1	1	2	3	2	3	7
P3	2	3	7(q)						
P4	1	1	2	2	2	3	4	7(q)	
P5	1	2	2	3	3	4(c)			
P6	1	1	2	2	7(q)				

Note: q = Quit prior to completion of maneuver
x = Quit between maneuvers
c = Run terminated due to computer failure

Head movement data. Analysis of Variance (ANOVA) tests were performed to determine if the extent of head movements in the yaw plane differed as a function of the four repetitions each of the Sawtooth and S-turn maneuvers performed in the AH-1S Cobra helicopter. The values used in the ANOVAs were the averages of each pilot's maximum head turns during each maneuver. The maxima were identified by inspection of the time histories of each pilot's head position during each flight by three independent judges. The data from one pilot was not used in this analysis because the raters were unable to reliably identify points of maximum head turn. The head movement data are therefore based on the data from five pilots. Head movements to the right and left were analyzed separately. The ANOVA's revealed that the maximum extent of head movements to the left during performance of the Sawtooth maneuver differed reliably among the four repetitions (df = 3,12; F = 7.23; $p < .005$). A Tukey pair-wise comparison test determined that differences were between the third and fourth (post-simulation) repetitions and the first (pre-simulation) execution of the Sawtooth maneuver.[6] Also, the extent of head movement during the third repetition differed reliably from the extent of head movement during the second repetition.

In all these cases the extent of head movements to the left were reduced during the post-simulation performance of the maneuver relative to the head movement extent before the simulated flights in the CSRDF. This is apparent from examination of Figure 4 which shows the changes in head movements to the left during performance of the Sawtooth maneuver. Figure 5 shows the extent of maximum head movements to the right. This figure is included simply to show the difference in the absolute magnitude of head movements to the left and right during the performance of the Sawtooth maneuver.

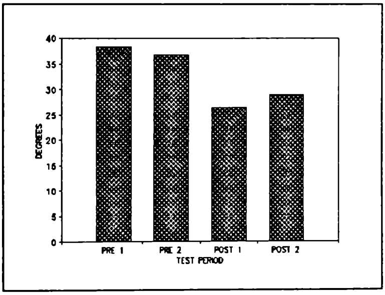


Figure 4. Extent of Head Turns to the Left of the Aircraft's Centerline During the Sawtooth Maneuver.

No other reliable differences in head movements were found for the Sawtooth and S-turn maneuvers performed in the aircraft.

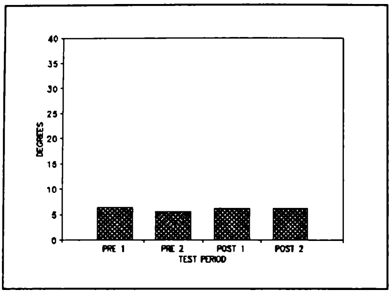


Figure 5. Extent of Head Turns to the Right of the Aircraft's Centerline During the Sawtooth Maneuver.

Maneuver Time. ANOVA tests were applied to the maneuver duration data to determine if the time to perform the Sawtooth and S-turn maneuvers in the AH-1S Cobra helicopter differed reliably as a function of maneuver repetition. The analyses were applied separately to the Sawtooth and S-turn maneuver data. Maneuver time data was collected for all six pilots. Time to perform the maneuvers was reliably different among the four repetitions for the Sawtooth maneuver (df= 3,15; F = 8.00; p < .002) and among the four repetitions of the S-turn maneuver (df = 3,15; F = 5.82; p < .008). Tukey pair-wise comparison tests revealed that for both the Sawtooth and the S-turn maneuvers the time to perform the third (post-simulation) repetition differed reliably from both the first and second (pre-simulation) executions of the maneuvers.

Figures 6 and 7 show the performance time data for the Sawtooth and S-turn maneuvers, respectively. It can be seen in these figures that the time to perform the post-simulation repetition is greater than the time to perform the same maneuver prior to the simulated flight in the CSRDF.

DISCUSSION

The hypothesis that simulator experiences of SIS are associated with reduced head movements in helicopter flight are supported by the head movement data. It must be pointed out that at the time of writing of this paper (June, 1992) data were not available from a control group of pilots who performed the same maneuvers in the aircraft on successive days but did not fly the CSRDF. Therefore, the post-simulation reduction in head movements during performance of the Sawtooth maneuver cannot strictly be attributed to experiences during the simulated flight; it may be that time alone is a sufficient explanation.

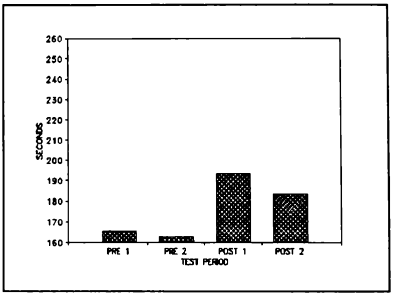


Figure 6. Mean Time Required to Complete Sawtooth Maneuver.

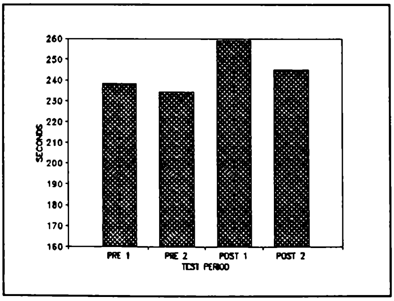


Figure 7. Mean Time Required to Complete S-Turn Maneuver.

For the Sawtooth maneuver, the likely reason that the head movements are larger to the left than to the right is that the lateral movement was 90 degrees to the left; moving to the right and forward, the helicopter advanced on a path 24.0 degrees to the right side of the next set of taxiway lights.

For both the Sawtooth and S-turn maneuvers the time to perform the maneuvers took significantly longer during the post-simulation runs than during the pre-simulation runs. The cause of these time differences can be attributed to a more-gentle performance of the maneuvers after the simulation to reduce the experience of physical and optical accelerative effects. Without the control group data, this explanation is also problematic.

These data suggest that the pilot's were attempting to control the adverse effects from the simulator experience by limiting both their head motions and the acceleration of the aircraft. Because of the strong SIS inducing effects of the maneuvers used, SIS appears to be the most likely cause of the changes noted in the post-simulation flight performance. That is, the pilots are reducing their head movements, and extending the maneuver times to ameliorate, or at least not aggravate the symptoms of SIS.

However, there is a real possibility that simple learning, not SIS, produced changes in subsequent performance in the helicopter. Although the data are not available at this time, preliminary examination of the simulator head movement and maneuver time data support this alternative. The raw data plots reveal a marked reduction in the extent of head movements while performing the Sawtooth and S-turn maneuvers in the simulator relative to performing them in the helicopter.

Moreover, the time to perform the maneuvers is longer in the simulator. The weight of the simulator helmet is about 2.2 kilograms, and a drag from the fiber-optic cables is experienced when yawing the head. Pilots may have responded to these characteristics by reducing their head movements. Because of their unfamiliarity with the control characteristics of the CSRDF, the pilots may have taken longer to perform the maneuvers. Consequently, when they flew the helicopter a short time later, the reduced head movements and longer maneuver times may be due to what was unintentionally learned in the simulator. Another study will be required to distinguish between SIS and learning as the causal agent (assuming it is not simply time) of these effects.

A remarkable outcome of this study was the high incidence and severity of the SIS reported by the pilots while performing the Sawtooth and S-turn maneuvers in the simulator, and the lack of adverse symptoms when the same maneuvers were performed in the AH-1S FLITE helicopter. In addition to the six subject pilots, three other pilots who flew these maneuvers prior to the formal data collection also reported severe symptoms of SIS and requested termination of the simulator session. In previous studies of SIS, the incidence of moderate to severe discomfort reported using the well-being scale usually ranges between 10 and 45 percent.

In the original conception of this study, it was assumed that the chase sequence in the middle of the simulator series of maneuvers would be the SIS inducing factor. The Sawtooth and S-turn maneuvers were designed to require side-to-side head movement but not fixation on a particular object for prolonged periods. Thus, how far the pilot's head turned would, to some degree, be up to the pilot. However, the pilots reported that the chase sequence was a form of relief and that it was the Sawtooth and S-turn maneuvers that induced SIS.

It was fortuitous that the Sawtooth and S-turn maneuvers turned out to be so nauseogenic. If these maneuvers, which we have dubbed the SIAHM maneuvers, also produce a similar incidence and severity of SIS in other simulators, they will serve as a valuable tool in the investigation of the physiological and behavioral bases of SIS. Moreover, discovery of what factors make the SIAHM maneuvers so potent as an inducer of SIS is worthy of investigation. The potency may come from the large number and amplitude of the head movements required, the frequent and large changes in optical flow, the low frequency of the maneuver motions, or some other factor.

Whatever the detailed explanations, this study has confirmed, subject to data from a control group, that simulator experiences can have adverse effects on piloting, at least shortly after the simulator experience. Also, this study confirms that SIS is a problem of some significance that should be addressed through research.

As a final point, it is worth bearing in mind that helmet mounted displays of synthesized and synthetic imagery are being developed for real aircraft. That is, in the cockpit, the display technologies for simulation and operational flight are converging. It may be that with all the good that may result from the transfer of simulation technologies to the operational environment, some of the problems, such as SIS, are likely to appear also.

REFERENCES

1. Uliano, KC; Kennedy, RS; & Lambert, ET. Asynchronous visual delays and the development of simulator sickness. Proceedings of the Human Factors 30th Annual Meeting. Dayton, OH: Human Factors Society, 1986.

2. McCauley, ME; Cook, AM; & Voorhees, JW. Recent proceedings of the NASA steering committee on simulator induced sickness. Report No. AIAA-91-2973-CP. American Institute of Aeronautics and Astronautics. Washington, D.C. 1991.

3. Henderson, BW. Simulators play key role in LHX contractor selection. Aviation Week & Space Technology. 1989 (27 November).

4. Lypaczewski, PA; Jones, AD; & Voorhees, JW. Simulation of an Advanced Scout Attack Helicopter for Crew Station Studies. Proceedings of the Interservice/Industry Training Systems Conference.

5. Haworth, L; Szoboszlay, Z; Shively, R; & Bick, F. AH-1s Communication Switch Integration Program. NASA Technical Memorandum 101053, USAAVSCOM Technical Report 88-A-009, March 1989.

6. Winer, BJ. Statistical Principles in Experimental Design. New York: McGraw Hill, 1962.

ACKNOWLEDGEMENTS

The authors wish to express thanks to MAJ L. Haworth, MAJ R. Seery, LTC R. Simmons and Mr. G. Tucker who served as safety pilots in this experiment, to Ms. C. Burston, Mr. D. Kennedy, and Mr. B. Wong for their assistance in collection of the data, and to Ms. G. Graham for her assistance with the analysis of the data.

The opinions expressed in this paper are those of the authors and do not necessarily reflect the official view of the U.S. Army or NASA.

AIAA-92-4137-CP

HELMET MOUNTED DISPLAY FLIGHT SYMBOLOGY RESEARCH

Loran A. Haworth[*] and Major Ronald E. Seery

U.S. Army Aeroflightdynamics Directorate (AFDD)
Crew Station Research and Development Branch
Ames Research Center, Moffett Field, California

Abstract

Screen/head stabilized MIL-STD 1295 Helmet Mounted Display (HMD) flight symbology as integrated on the Apache helicopter was compared to world referenced/stabilized flight symbology. Simulation test results indicate that pilots perform significantly better using world stabilized conformal attitude symbology. They were accurate to an average of 1/2 degree at estimating terrain relief and aerial target locations. Pilots were able to take advantage of world referenced symbology due to the unique features of HMD that allow the pilot to visually use the symbology at extreme azimuth and elevation off-axis angles. World stabilized conformal symbology was also perferred while performing contour flight tasks. They reported that the use of climb-dive-marker during contour flight greatly reduced pilot work load under conditions tested. Cyclic input errors occured when using both MIL-STD 1295 hover symbology and test symbology indicating that a better approach for depicting hover symbology is warranted. The magnitude of cyclic input and spatial estimation errors increased as the off-axis viewing angle became larger.

Nomenclature

AC	Acceleration Cue
ADOCS	Advanced Digital Optical Control System
AFDD	Army Aeroflightdynamics Directorate
AGL	Above Ground Level
AH-64	U.S. Army Apache Helicopter
CDM	Climb Dive Marker
ARC	Ames Research Center
BWR	Bedford Workload Rating
CDM	Climb Dive Marker
CHL	Conformal Horizon Line
CRB	Compass Rose Box
CSRDF	Crew Station Res. & Dev. Facility
FCR	Fixed Compass Rose
FHR	Fixed Horizon Line
FLIR	Forward Looking Infrared
FOV	Field Of View
FSWG	Flight Symbology Working Group
HD	Heading Digits (Head Azimuth)
HUD	Head-Up-Display
HMD	Helmet Mounted Display
HQR	Handling Qualities Rating
IR	Infrared
KIAS	Knots Indicated Airspeed
KTAS	Knots True Airspeed
LED	Light Emitting Diode
LOS	Line of Sight
MCR	Moving Compass Rose (Head Azimuth)
OHT	Optical Head Tracker
PL	Pitch Line/bar
PNVS	Pilot Night Vision System
TLX	Task Workload Index
USA	United States Army
VV	Velocity Vector

Definitions

Head Stabilized/Referenced Symbology – Symbology maintains the same position independent of head and eye movement. Also referred to as Screen-Stabilized Symbology.

[*] Research Psychologist and Engineering Test Pilot. Member AIAA.

Aircraft Stabilized/Referenced Symbology – Symbols maintain virtual position relative to the aircraft.

World/Stabilized/Referenced Symbology – Symbols remain in position relative to the world/earth.

Conformal Stabilized/Referenced Symbology – Location, dynamics and behavior of the symbol corresponds to visual/sensor image. e.g. horizon line on the world horizon. Also called Contact Analog.

Background

The investigation examined the stabilization of flight symbologies for head-tracked HMD systems which are described in Military Standard 1295 (MIL-STD-1295), "Human Factors Engineering Design Criteria for Helicopter Cockpit Electro-Optical Display Symbology" and are integrated on the U.S. Army's (USA's) AH-64 Apache helicopter. In addition, head-stabilized flight symbology as found on the Apache helicopter and in MIL-STD-1295 was directly compared to world referenced flight symbology during simulation tests. Stabilization of the horizon line, velocity vector, acceleration cue, and compass indications were examined. Additional flight information to include pitch lines/bars and climb-dive angle indications were also studied. The five week flight simulation was performed using the Crew Station Research and Development Facility (CSRDF) at the NASA Ames Research Center (ARC), Moffett Field, California. Six U.S. Army and two United Kingdom helicopter pilots participated in the study. Both objective and subjective measures were obtained as pilots performed hover, low-level flight, and up-and-away flight tasks. The flight symbology investigation conducted by AFDD was coordinated and supported by The Tri-service Flight Symbology Working Group (FSWG), The Technology Cooperation Program – Subgroup H, Technical Panel 6, "Helicopter Aerodynamics and Man Machine Integration" (TTCP-HTP6), and NASA ARC.

Introduction

The USA's AH-64 Apache helicopter is presently the only production aircraft in the United States that incorporates a HMD with a integrated flight symbology graphics generator (Fig. 1). The HMD and symbology generator are part of the Apache Pilot Night Vision System (PNVS). The PNVS presents a head-steered Infrared (IR) image overlaid with flight symbology to the pilot (Fig. 2), via a monocular HMD called the Integrated Helmet and Display Sighting System (IHADSS). Flight symbology displayed on the IHADSS HMD is referred to as screen or head fixed symbology since the location of the symbology on the display surface is independent of head position. AH-64 Apache electro-optic flight symbology is described in MIL-STD-1295.[1] AH-64 symbology and nomenclature is presented in Figs. 3 and 4.

Essentially the Apache HMD is a miniature fixed wing Head-Up-Display (HUD) that attaches to the pilot's helmet. Unlike the fixed HUD, the HMD can be viewed when the head is moved from the longitudinal axis (off-axis) of the aircraft. The HMD is visually coupled with the Forward Looking Infrared (FLIR) in azimuth and elevation . The pilotage sensor follows the pilot's head movement which enables the pilot to direct the FLIR off-axis while viewing both flight symbology and sensor imagery. However, when the pilot moves his head to an off-axis position, screen/head stabilized spatial flight symbology presented via the IHADSS no longer corresponds with the IR visual image presented by the sensor. That is, attitude and directional information such as the horizon line and velocity vector no longer coincide with the viewed image (Fig. 5). The situation is further complicated since the FLIR image is not head-roll compensated and the symbology developed for hover and position cueing is visually more appropriate for a head-down-display orientation. All the above makes it cumbersome for the pilot to integrate visual sensor information and symbolic spatial information at the same time.

The USA's requirement to operate military helicopters in low-altitude and night operations, during Operation Desert Storm, illustrated the continued need for HMD devices that incorporate both sensor information and flight data. A dual ocular head tracked HMD with integrated flight symbology and sensor information is planned for future systems such as the USA's RAH-66 Comanche, the Navy's V-22 Osprey, and the Air Force's F-16.

The authors and the FSWG predicted that world conformal stabilized flight symbology would improve the pilot's situational awareness and reduce workload when compared to head/screen referenced MIL-STD-1295 AH-64 symbology. World referenced conformal symbology remains fixed in the world as the pilot moves his head. Pilots using the world coordinate system would not have to mentally translate head fixed symbology to the world system as required with head/screen stabilized symbology. Flight symbology should appear fused and conformal with the world IR/visual image. For example, at low altitudes world stabilized horizon lines appears to overlay the distant visual world horizon and directional information such as the Velocity Vector (VV) would more closely indicate the appropriate direction of travel.

Purpose

The purpose of the simulation study was to compare HMD flight symbology stabilization/ reference algorithms and to use the investigation as a basis for recommending prefered symbology for advanced rotorcraft. Simulation investigations of symbology will provide design logic and guidance for next generation/future growth systems leading to improved pilotage performance, flight safety, and reduced workload for helicopter aircrews.

Scope

The investigation was limited to generic issues concerned with reference and stabilization of groups and categories of flight symbology as described in MIL-STD-1295 and integrated on the AH-64 Apache helicopter. Specifically, the scope of the investigation was the comparison of screen/head stabilized symbology to world stabilized conformal symbology. Symbolic icons representing flight data were not manipulated for comparison during this limited simulation. Where possible MIL-STD-1295 icons were retained for a direct comparision. Stabilization of the horizon line, PLs, CDM, magnetic heading indications, VV and Acceleration Cue (AC) were examined. Although weapon and status information is important and must be integrated within the HMD, only flight symbology stabilization was addressed during this investigation.

Method

Facility

The investigation was conducted in AFDD's fixed based CSRDF's flight simulator[2-4] in September 1991. The visual display was presented using the fiber optic HMD depicted in Fig. 6. The Field-Of-View (FOV) of the HMD was mechanically limited to 30 degrees vertically and 52 degrees horizontally to represent the FOV proposed for the RAH-66 Comanche Helicopter HMD. The General Electric Compu-scene IV data base served as the visual data base for the experiment. Both IR and daylight visual tables were used during the course of the investigation. Raster flight symbology was generated by a 2300 Turbo Silicon Graphics IRIS and presented within the 25 X 19 degree high resolution inset of the HMD. Aural cueing was employed throughout the simulation for rotors, engines, and transmission. A VAX 8650 computer generated the mathematical aircraft model and another Silicon Graphics 2300 Turbo generated the head-down cockpit displays.

Control Laws

The flight control laws used for this simulation were based on the Advanced Digital Optical Control System (ADOCS)[5] with slight modifications for CAE side stick displacement controllers. Basically, the control system was a model following system with feed forward shaping and feedback compensation. The side-stick cockpit controllers were configured as a conventional 2 + 1 + 1 system. An attitude command/ velocity stabilization control system was used for pitch and roll. Turn coordination and heading hold were always available during flight.

Aircraft Math Model

Simulation of the flight vehicle (UH-60A Blackhawk) was provided by a generic single main rotor helicopter math model. The math model is described in Ref. 6.

Experimental Variable

Two sets of experimental variables were manipulated for the investigation. They were task condition and flight symbology presentation. A total of nine flight symbology presentations and five tasks were examined during the experiment. Task conditions and symbology presentations are described later.

Subjective Data Collection Techniques

Three subjective ratings were used during the investigations: (1) the Cooper-Harper Handling Qualities Rating (HQR) scale[7,8] for determination of aircraft handling qualities; (2) The NASA Task Load Index (TLX) that divides workload into several categories[9-11]; and (3) the Bedford Workload Rating (BWR) scale for determination of total pilot workload.[12] Higher BWR (1 - 10 range) and TLX ratings (1 - 100 range) indicate higher pilot workload. Higher HQRs (1 - 10 range) indicate higher workload and degraded performance. HQRs were only obtained for position recovery and contour flight tasks since both spatial awareness tasks were not handling quality tasks. In addition, post simulation questionnaire data was collected at the conclusion of the experiment.

Data Reduction

Standard handling qualities data and head tracked information were recorded on computer disk packs and stored for later data reduction. Objective data analysis was performed using the Statistical Analysis System version 5.0 on a microVAX II computer. The general linear models procedure was the primary analysis tool. The Student-Newman-Kreuls multiple range test was performed on all main effects.

Pilots

Six U.S. Army helicopter pilots from the Test and Evaluation Command (TECOM) and two United Kingdom helicopter pilots from Farnborough, England participated in the experiment. Four of the pilots were qualified in the AH-64 Apache helicopter equipped with the IHADSS HMD. Six of the eight were experimental test pilots. Pilot experience is summarized in Fig. 7.

Experimental Design

A within subjects repeated measures experimental design was performed. The design replicated each trial within each subject/task combination. Performance times for tasks were counter-balanced between pilots to prevent biases due to learning effects and/or pilot fatigue. Trial presentations between tasks were randomized to reduce the effect of learning and/or pilot fatigue.

Flight Tasks

The five task and flight conditions, shown in Fig. 8 were simulated. TLX, BWR, HQR ratings and pilot comments were gathered at the conclusion of each task, where applicable.

Position Recovery Task Description

The purpose of this task was to determine the pilot's ability to recognize his hovering flight condition and to use this information to recover to a hover position. Specifically the pilot was required to recover the helicopter from hovering flight to a stabilized hover position as rapidly as possible while visually tracking a stationary target both on-axis and off-axis from the aircraft centerline. This approximates an actual situation where the pilot encounters a visual target during hovering flight and brings the helicopter to a hover while continuing to monitor the target. The situation required the pilot to use visual information presented by the IR image and flight symbology to aid in recovery of the aircraft while forcing the pilot to maintain his head off-axis. Noise was added to the IR image to approximate a standard AH-64 IR image as determined by experienced AH-64 pilots.

Targets were presented as white hot objects on the data base. The target was presented at one of five different azimuth positions. Once the pilot sighted the target, flight symbology was presented and the simulated aircraft was released to the pilot's control at seven knots ground speed in one of four preset hover directions. The pilot was instructed to use the cyclic to recover the aircraft to a hover position while maintaining his head aligned within 10 degrees of the presented target. Task standards required an initial recovery control input within two seconds and a stabilized hover position within 12 seconds.

Low Altitude Spatial Awareness Estimation Task Description

The objective of this task was to provide a set of simulated conditions requiring the pilot to use the presented flight symbology for spatial estimation tasks. The task required the pilot to estimate the vertical and directional angles to selected terrain in relation to the simulated aircraft. Spatial estimates were made from three low altitude locations. Visibility was limited to two miles without a visible horizon. The task simulated visual situations encountered during low level flight in varying terrain where the actual horizon could not be viewed. This static visual task was performed to first provide investigators insight into the pilot's ability to accomplish visual estimations accurately using presented symbology without introducing additional variable effects as a result of piloting ability, aircraft dynamics and additional workload.

Contour Flight Task Description

The task demonstrated the pilot's ability to use dynamic flight symbology in performing contour flight under test conditions. The pilot was required to perform contour flight at 50 Knots Indicated Airspeed (KIAS) and 30 feet Above Ground Level (AGL). The pilot flew a straight flight course for five minutes and S turns for two minutes. The flight control task was one primarily of controlling altitude with the collective over hilly terrain. The contour flight task required the pilot to use both visual information collected from the IR image and flight symbology under dynamic conditions.

Up and Away Spatial Awareness Estimation Task Description

The purpose of this task was the same as the low altitude spatial awareness task but under up and away visual conditions as described below. The task required the pilot to estimate the vertical location and azimuth of a target aircraft in reference to the ownship location while in flight at 200 feet AGL and 60 KIAS. Target aircraft were visually presented in a welded wing format (constant angular and distance location). Targets were only presented for five seconds once the pilot visually sighted the target. Aircraft locations were varied in both azimuth and elevation relative to the ownship. Pilots were asked to estimate the vertical angle and azimuth to the target aircraft. This task demonstrated the pilot's ability to spatially estimate the location of aerial targets using presented symbology and visual imagery.

Spatial Orientation Estimates Task Description (without symbology)

The objective of this task was to examine the pilot's ability to estimate head orientation at various off-axis angles and over a time period. This task required the pilot to turn his head to instructed head azimuth positions without symbology. The task was performed by each pilot at the beginning and end of each simulation period.

Flight Symbology Conditions

General. The test matrix is presented as Fig. 9. Further explanations regarding both symbology presentations for each flight task are provided below. Symbology nomenclature is provided.

Position recovery symbology. Pilots performed position recovery tasks using MIL-STD-1295 AH-64 screen referenced VV and Acceleration Cue (AC) symbology (Fig. 10) found in the hover mode display. Pilots were also required to perform the same task using world referenced VV and AC AH-64 symbology (Fig. 11). Direct comparisons were made between the two stabilization methods.

The VV and AC symbology originates from the center of the IHADSS and indicates the direction and magnitude of the ownship's horizontal velocity and acceleration. The head/screen fixed symbology remained independent of head movement, however, the world referenced symbology rotated around the point of origin as the pilot moved his head in azimuth. This allowed the world stabilized VV and AC to visually point in the direction of aircraft movement. A small arrow shaped icon was added to the world referenced symbology set to indicate the longitudinal axis of the aircraft.

AH-64 VV and AC icons move on a mostly vertical display depending on head orientation, however, these icons represent horizontal movement over the ground. This forces the pilot to mentally transpose symbolic information presented vertically to a horizontal plane. Stated another way , the VV vector line originating from the center of the display and pointing up from the center of the screen does not indicate upward movement, but movement across the ground in the horizontal plane. This situation exists with both symbology sets but to a lesser degree with the world stabilized VV and AC.

Low altitude spatial awareness estimation symbology. Elevation comparisons were made between the standard AH-64 Fixed Horizon Line (FHR) (Fig. 12) and AH-64 symbology modified with a world Conformal Horizon Line (CHL) (Fig. 13). World stabilized PLs were displayed with the CHL icon to provide additional information for elevation and attitude estimations.

Pilots performed head azimuth estimations using four separate indications of azimuth (Fig. 14):
1) Standard AH-64 Fixed Compass Rose (FCR),
2) Moving Compass Rose (MCR) that rotated with head azimuth changes, 3) Compass Rose Box (CRB) that indicated the head azimuth direction with a box on the FCR, and 4) Heading Digits (HD) that displayed a digital presentation of head azimuth direction above the fixed compass rose.

Contour flight symbology. The pilot maintained contour flight using the three flight symbology conditions indicated in Fig. 15. The conditions were 1) AH-64 cruise symbology with screen fixed Apache FHL and FCR, 2) AH-64 symbology modified with a world-conformal referenced CHL, PLs, FCR and aircraft stabilized gull wings, and 3) AH-64 symbology with a world-conformal referenced CHL, PLs, Climb-Dive Marker (CDM), FCR and wings. The CDM (Fig. 15) visually represents the aircraft impact point and provides predictive information for maintaining the flight path. A gull wing icon was added to replace the aircraft referenced Line-Of-Sight (LOS) reticle fоr contour flight.

Up and away spatial awareness estimation symbology. Pilots performed elevation and azimuth estimations of a target aircraft relative to the ownship under five symbology conditions. Conditions are identified in Fig. 9. In addition to the four symbology sets examined during the low altitude spatial estimation task, an aircraft stabilized HUD was also provided as a test condition.

Results and Discussion

Position Recovery

The percentage of incorrect cyclic control responses (Fig. 16) was surprisingly high during hover recovery tasks using the MIL-STD-1295 hover symbology and the world stabilized AC and VV symbology set. An incorrect control response was graded as one where the pilot did not move the cyclic in the correct direction (±45 degrees) to reduce the hover velocity. The data indicates a high level of pilot-reported difficulty interpreting both hover symbology sets, especially during off-axis viewing at angles greater than 45 degrees. The higher error rate associated with the modified symbology are due to the learning effect for AH-64 pilots. While learning effect may explain some amount of error, neither symbology set was considered satisfactory by the authors for allowing the pilots to predict the correct initial cyclic control input for hover recovery under conditions tested.

Several additional reasons reported and observed may help to explain the high percentage of initial cyclic input errors. These include: 1) errors estimating head position, 2) head-down display icons presented in the vertical plane, 3) body position in relation to the side stick controller during off-axis sighting and 4) screen referenced symbology. Conditions that lead to incorrect control inputs should be explored further since it is indicative of confusion associated with interpretation of hover symbology.

During the debriefings, pilots reported that cyclic applications to correct for aircraft drift were often applied in the wrong direction when the head was turned more than 30 degrees off-axis using MIL-STD-1295 symbology. AH-64 pilots also reported negative habit transfer from the AH-64 when using the modified MIL-STD-1295 symbology set leading to increased cyclic input errors. The level of visual clutter in the hover mode was rated as low for both symbology sets and the mental workload associated with use of both sets was reported as high. Pilots often complained about the "mental arithmetic" that had to be performed to translate hover symbology to the correct control action. The ability to determine horizontal drift rapidly was rated slightly better using the modified symbology, however pilots indicated slightly more confusion as to the proper direction of control of movement. Pilots indicated that they used symbology more than scene content (75% symbology, 25% scene content) for the hover recovery task. AH-64 pilots indicated the same trend was true for actual flight. Pilots also stated that neither symbology set was satisfactory and a new approach should be examined.

Objective data indicated that target location significantly ($p < 0.05$) effected performance measures shown in Fig. 17. Essentially the time to recovery and the distance traveled to recovery increased as the off-axis target angle increased especially for 90 and 45 degrees left. Also significant differences between the two symbology sets were demonstrated on "time until stable hover was first achieved" and "elapsed trial time." Pilots using the MIL-STD-1295 hover mode symbology set were significantly faster in achieving hover and end of task when compared to the AH-64 symbology modified world stabilized VV and AC symbology. The cumulative distance traveled was significantly less with MIL-STD-1295 symbology. Faster recovery time and less cumulative distance using the MIL-STD-1295 symbology is partly attributed to fact that four of the pilots were experienced using the AH-64 symbology. Results were based on 605 objective observations.

Low Altitude Spatial Awareness

Pilots performed azimuth estimations significantly ($p < 0.05$) better using MCR, CRB, or HD in comparison to the standard AH-64 FCR. Average errors of approximately 1/2 degree were achieved using the HD shown in Fig. 18. Pilots predicted that the MCR could be confusing when the aircraft was turning; however this could not be quantified. Pilots also reported that the CRB was somewhat difficult to interpret when the box partially overlaid the numbers on the FCR. Pilots overwhelmingly felt that the HD, as presented, led to reduced pilot workload and increased performance when compared to the MCR, AH-64 FCR and the CRB. Results were based on 694 observations.

In addition to increasing general spatial awareness, the ability to accurately interpret the azimuth to a location or target could be of great tactical value when accurate estimations of threat and friendly personnel must be relayed for target handoff. This could also reduce the reported occasions where the pilot would physically turn the aircraft toward the target to obtain accurate azimuth information thus telegraphing possible intentions to the threat forces.

Elevation estimations using world stabilized CHL and PLs were statistically significantly ($p < 0.05$) better when compared to the standard AH-64 symbology set. Mean errors of less than a 1/2 degree were achieved as shown Fig. 17. Pilots indicated that they were able to determine terrain relief more accurately using a world CHL and PLs in comparison to the Apache screen fixed FHL.

The lowest workload was associated with the AH-64 symbology set with the world stabilized/ referenced CHL, PLs, HD and FCR for the low altitude elevation and azimuth estimation task. It was also observed that the physical closeness of the display to the eye combined with world stabilized attitude information allowed the pilot to interpret the spatial relationship of objects more accurately since the visual world was stabilized and calibrated with PLs, heading indicators and the CHL.

With an accurate indication of terrain relief associated with world stabilized symbology, pilots will perform better with a reduced pilot workload under low altitude flight conditions and during slope landings. This symbolic information also leads to increased flight safety for nap-of-the-earth and/or contour flight since it is more likely that the pilot would avoid ground strikes.

Contour Flight

No significant differences were observed between symbology sets for the contour flight task. Forty six observations were made for contour flight. Subjective ratings on the HQR, BWR, and TLX were approximately the same for each symbology set. Radar altitude symbology was present in each symbology set. Radar altitude information combined with the world visual presentation that was present in each condition, the limited number of observations, and the fact that pilots can compensate for conditions over shadowed any difference due to symbologies for the contour flight task.

Pilots indicated on end of test questionnaires that AH-64 symbology with world referenced CHL, PL, and CDM provided a better indication for correct controling movements. It was also easier to interpret combined FLIR and symbolic information when using the world referenced symbology. In addition, the CDM was centrally located in the display which allowed the pilot to maintain better visual contact with the FLIR imagery since the pilot did not have to look at the the right side of the symbology display as often for altitude information. When asked to rate the importance of the CHL, PLs and CDM separately for helping in reducing workload, estimating terrain variances, and increasing pilot performance during contour flight, the pilots rated all as being valuable, with the CDM clearly rated as the most valuable during contour flight.

Since the CDM presents a visual flight path marker, the pilots were able to set the predictive CDM symbol using the collective at a selected visual height above the terrain. When traversing over hilly terrain, the pilots were also able to visually follow the CDM at high angles of elevation and depression so the CDM could be visually placed above high terrain or down a slope. This capability is unique to the HMD since the CDM movement and usefulness is limited on a HUD display. The predictive nature of the CDM also provides the pilot with a limited indication of whether the aircraft has enough energy/ power to clear terrain above the aircraft since the marker symbol overlays the visual impact point. Researchers noted less frequent collective control movements when using the predictive CDM. Further investigations should be conducted examining the use of the CDM in HMD for low altitude flight.

Up and Away Spatial Estimation

As predicted pilots performed elevation estimations significantly better ($p < 0.05$) using symbology with a world referenced CHL and PLs. As cited during the low altitude azimuth estimation tasks, the addition of a MCR, CRB or HD significantly increased azimuth accuracies shown in Fig. 19.

Azimuth and elevation estimation errors generally increased as aerial targets were presented at increased off-axis angles from the nose of the aircraft. Elevation estimation errors were also significantly ($p < 0.05$) affected with increased off-angle/axis target presentations. As found in the Low Altitude Spatial Awareness Task the symbology set with the world stabilized CHL, PL, HD and FCR received the lowest workload ratings. The fixed HUD display received the highest.

The increased spatial awareness of the ownship relationship to another aircraft in flight has several potential benefits. These include better energy management for air-to-air targeting, increased safety during night formation aerial join ups, increased safety for formation flying and increased spatial awareness of the relationship of the aircraft to the world. While not specifically measured it inferred that world stabilized symbology 1) would also reduce the need to use head down aircraft instrumentation for instrument flight conditions, 2) would aid in unusual aircraft attitude recoveries, and 3) would significantly increase spatial information for aerobatic flight that may be encountered during air-to-air engagements. Further, up and away investigations should be conducted to study world stabilized HMD flight symbology for air-to-air, instrument flight, and unusual attitude recovery tasks.

Spatial Estimation Task

Significant differences ($p< 0.05$) in estimation accuracies were obtained during the spatial estimation task (Fig. 20). The greatest estimation error was recorded at –30 degrees and the least at 0 degrees. Pilots had also commented during the Hover Position Recovery Task that more recovery errors were made at off-axis angles beyond 30 degrees. Pilots reported difficulty in estimating head relationship to the aircraft during trials, and it was suspected that this would become worse over time. However, the head estimation error did not significantly increase with trial time or time of the day as expected. While this test did not demonstrate increased problems with head orientation over time, further time-dependent investigations should be conducted.

Concluding Remarks

Position Recovery Task

A surprising number of incorrect initial cyclic input errors were made while performing position recovery tasks. This suggests a high level of pilot difficulty in interpreting the MIL-STD-1295 hover symbology, especially when viewed off-axis. Pilots reported difficulty interpreting hover symbology when viewing off-axis and complained about the "mental arithmetic" required to calculate the correct control response based on the hover symbology. Neither symbology set examined under test conditions was considered satisfactory for rapidly determining which direction to initially move the cyclic control stick. While the position hover task could be performed, both pilots and investigators suggest that other approaches for hover symbology presentations should be examined.

Low Altitude Spatial Estimation

Pilots performed elevation and azimuth estimation tasks significantly better using world-stabilized conformal flight symbology. Pilots were able to accurately determine terrain relief and the magnetic direction of the selected terrain. With a more accurate indication of terrain relief it is predicted that flight safety will be enhanced and pilot workload will be reduced under low altitude flight conditions Increased azimuth accuracies will increase general spatial awareness and will have some tactical value when providing directional information in relation to the ownship.

Contour Flight

No significant differences were recorded between symbology sets during contour flight. However pilots indicated that world-referenced conformal symbology reduced pilot workload. The CDM was clearly rated as being the the greatest contribution to reduced pilot workload and increasing performance. Since the CDM was centrally located in the display, pilots found it easier to view the FLIR imagery without having to look toward the edge of the display to view the analog and digital readout of altitude. Pilots reported they could interpret combined FLIR imagery and symbolic information easier when world stabilized conformal symbology was provided. Continued investigations examining world stabilized conformal symbology for low altitude flight are recommended.

Up and Away Spatial Estimation

As with low altitude estimation tasks, pilots performed significantly better in providing elevation and azimuth estimates of aerial targets when using world stabilized information. The symbology set that provided world stabilized CHL, PLs, HD and FCR symbology, received the lowest workload rating. The increased spatial awareness associated with the relationship of one aircraft to another has several potential benefits. These included better energy management for air-to-air targeting, increased spatial information for aerial join-ups at night, increased flight safety for formation flight and increased awareness of attitude information.

Spatial Estimation Task

Significant differences in off-axis estimations were found during the spatial estimation task. The greatest estimation error was recorded at –30 and the least at 0 degrees. This identifies pilot difficulty in estimating head orientation during off-axis viewing under conditions tested and may amplify the need for aircraft stabilized symbology. Pilots have reported losing precise head orientation with dual optic displays since pilots may lose visual contact with the cockpit.

Summary

In summary world stabilized conformal flight symbology provides accurate spatial information for low altitude flight and for determining aerial target location. Pilots were able to determine terrain relief and aerial target location within 1/2 degree accuracy. They were able to take advantage of the unique features of HMD that allow the pilot to visually use the world stabilized and virtual symbology at large azimuth and elevation angles off aircraft centerline. During contour flight pilots indicated their preference for the climb dive marker symbology that provides predictive flight path information. Data obtained during position recovery task indicated a surprising number of initial cyclic input errors indicating pilot confusion when interpering hover symbology. The number of cyclic input errors and estimation errors increased during off-axis viewing. This lead both pilots and investigators to conclude that the MIL-STD-1295 symbology and test symbology was confusing and not satisfactory for position recovery tasks. It is predicted that world stabilized flight information should enhance instrument flight, multi-ship night operations, unusual attitude recovery, acrobatic flight, terrain relief awareness, flight path prediction, and other flight regimes where spatial and attitude awareness is important. Further simulation and flight investigations are warranted.

Acknowledgment

The Authors wish to thank Mr. Millard Edgerton, Carla Burston, Perry Meade, Steve Rogers and Kinga Perlaki for their research support.

References

[1] Department of Defense (1984). Military Standard 1295A: Human factors engineering design criteria for helicopter cockpit electro-optical display symbology. Washington, D.C.: U.S. Government Printing Office.

[2] Henderson, B. W., "Simulators Play Key Role in LHX Contractor Selection," Aviation Week and Space Technology, Nov. 27, 1989.

[3] Lypaczewski, P. A., Jones, A. D., and Voorhees, J. W., "Simulation of an Advanced Scout Attack Helicopter for Crew Station Studies.

[4] Haworth, L. A. and Bucker, N., "Helmet Mounted Display Systems for Helicopter Simulation," Published in the Proceedings of the Human Factors Society 33rd Annual Meeting, Vol. 1, pp. 86-90, Oct 1989.

[5] Landis, K. H. and Glusman S. I., "Development of ADOCS Controllers and Control Laws," Vol I-III, USAVSCOM TR 84-A-7, March 1987.

[6] Howlett, J. J., "UH-60A Blackhawk, Engineering Simulation Program: Vol. 1 – Mathematical Model," NASA CR-166309.

[7] Cooper, G. E. and Harper, R. P., "The Use of Pilot Ratings in the Evaluatin of Aircraft Handling Qualities," NASA TN D-5153, Apr. 1969.

[8] Mudd, S. A. "The Treatment of Handling-Qualities Ratings Data," Human Factors, pp. 321-330, 1969.

[9] National Aeronautics and Space Administration. "NASA Task Load Index (TLX) ," Human Performance Research Group, NASA Ames Research Center. Moffett Field, CA. Undated.

[10] Vidulich, M. A. and Tsang, P. S., "Assessing subjective workload assessment: A Comparison of SWAT and NASA-Bipolar methods." Proceedings of the Human Factors Society 29th Annual Meeting, pp. 71-75.. Santa Monica, CA: Human Factors Society, 1985.

[11] Haworth, L. A., Bivens, C. C., and Shively, R. J., "An investigation of single-piloted advanced cockpit and control configuations for nap-of-the-earth helicopter combat mission tasks," Proceedings of the 1986 Meeting of the American Helicopter Society, pp. 657-672, Washington, D.C., 1986.

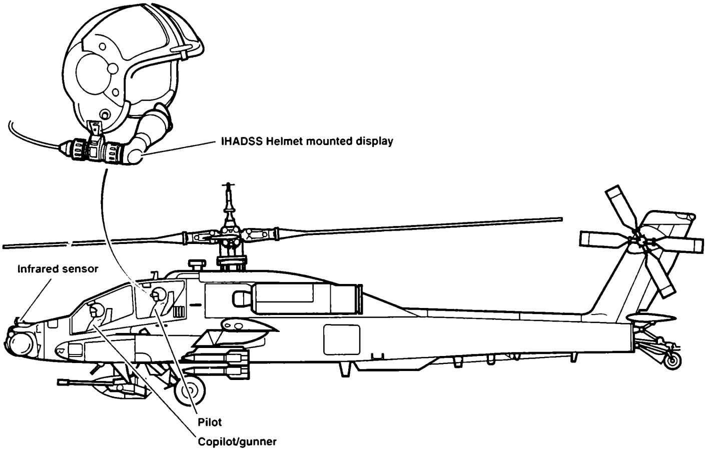


Fig. 1 AH-64 Apache Helicopter.

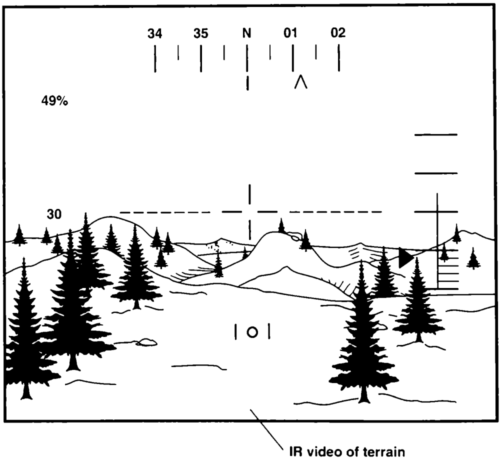


Fig. 2 IR image overlaid with flight symbology.

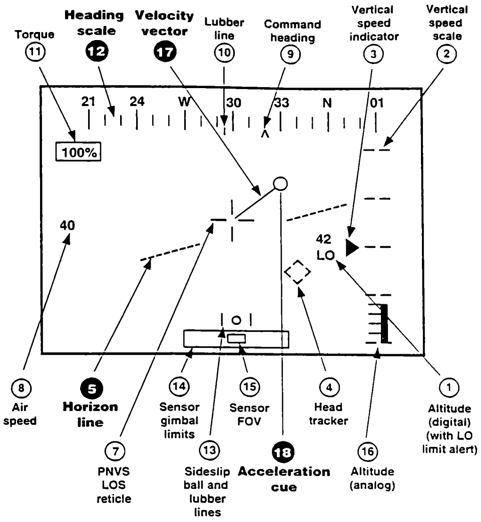
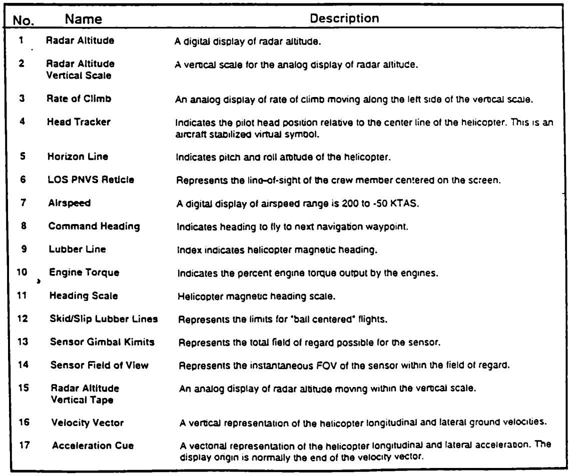


Fig. 3 AH-64 helmet mounted display symbology.



No.	Name	Description
1	Radar Altitude	A digital display of radar altitude.
2	Radar Altitude Vertical Scale	A vertical scale for the analog display of radar altitude.
3	Rate of Climb	An analog display of rate of climb moving along the left side of the vertical scale.
4	Head Tracker	Indicates the pilot head position relative to the center line of the helicopter. This is an aircraft stabilized virtual symbol.
5	Horizon Line	Indicates pitch and roll attitude of the helicopter.
6	LOS PNVS Reticle	Represents the line-of-sight of the crew member centered on the screen.
7	Airspeed	A digital display of airspeed range is 200 to -50 KTAS.
8	Command Heading	Indicates heading to fly to next navigation waypoint.
9	Lubber Line	Index indicates helicopter magnetic heading.
10	Engine Torque	Indicates the percent engine torque output by the engines.
11	Heading Scale	Helicopter magnetic heading scale.
12	Skid/Slip Lubber Lines	Represents the limits for "ball centered" flights.
13	Sensor Gimbal Kimits	Represents the total field of regard possible for the sensor.
14	Sensor Field of View	Represents the instantaneous FOV of the sensor within the field of regard.
15	Radar Altitude Vertical Tape	An analog display of radar altitude moving within the vertical scale.
16	Velocity Vector	A vertical representation of the helicopter longitudinal and lateral ground velocities.
17	Acceleration Cue	A vectorial representation of the helicopter longitudinal and lateral acceleration. The display origin is normally the end of the velocity vector.

Fig. 4 AH-64 symbology nomenclature.

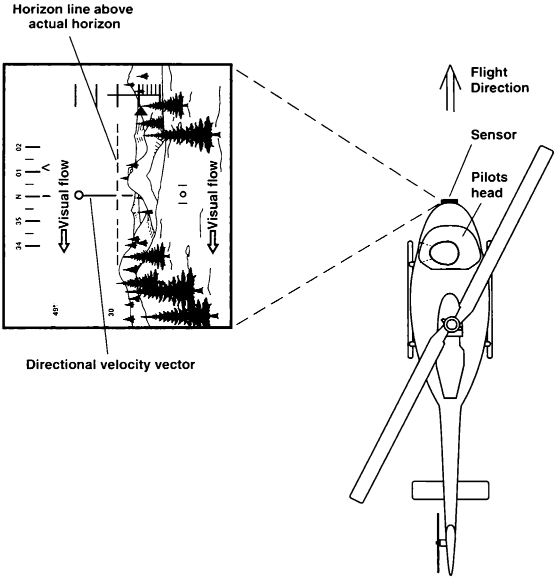


Fig. 5 Pilot viewing left.

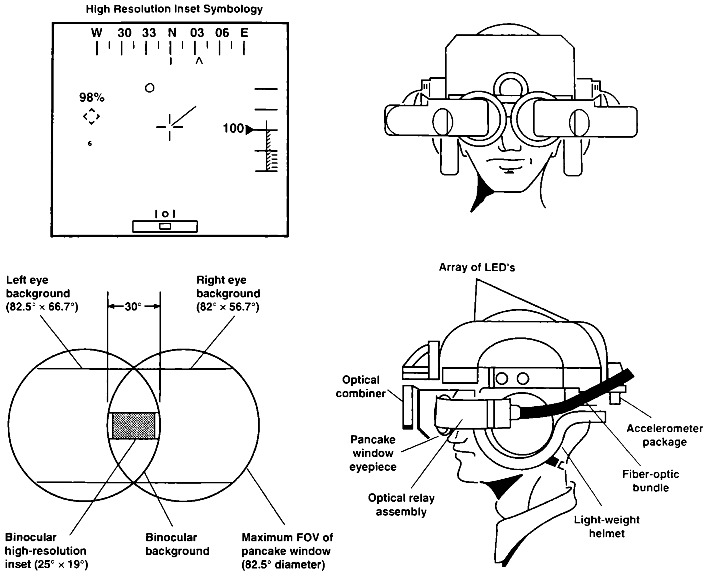

High Resolution Inset Symbology
W 30 33 N 03 06 E
98%
6
100
Left eye background (82.5° × 66.7°)
30°
Right eye background (82° × 56.7°)
Binocular high-resolution inset (25° × 19°)
Binocular background
Maximum FOV of pancake window (82.5° diameter)
Array of LED's
Optical combiner
Pancake window eyepiece
Optical relay assembly
Accelerometer package
Fiber-optic bundle
Light-weight helmet

Fig. 6 Fiber optic helmet mounted display.

Pilot experience

Pilot	HMD			NVG			Total hours	
	Simulation (hours)	Flight (hours)700	Aircraft	Simulation (hours)	Flight (hours)700	Aircraft	Helicopter	Fixed wing
1	150	700	AH-64 V-22	0	80	AH-1	6500	100
2	70	600	AH-64	20	250	AH-1F	2600	55
3	0	0	N/A	0	60	UH-60	2000	1500
4	30	350	AH-64	0	500	OH-58	2800	300
5	20	0	N/A	0	50	OH-58 UH-60	2500	0
6	200	400	AH-64	0	10	UH-60	5200	500
7	8	10	Lynx Jetstream AH-64	0	40	Seaking Puma Lynx	2200	300
8	15	5	AH-64 Res/Sim	0	70	Gazelle Puma Lynx	1700	70

Fig. 7 Pilot experience.

Flight task	Flight conditions							
	Visual				Flight control			
	Visibility (Miles)	Imagery		Data base content	ATT CMD VEL HOLD	POS HOLD	ALT HOLD	
		IR	Day				RAD	BARO
Position recovery	1	*		Flat terrain w/o visible horizon	*			*
Low alt spatial awareness	2	*		Rolling terrain w/o visible horizon	*	*		*
Contour flight	2	*		Rolling terrain w/o visible horizon	*	NA	*	
Up and away spatial awareness	7		*	Rolling terrain with visible horizon	*	NA		*
Spatial orientation	7	*		Flat terrain with visible horizon	NA	NA	NA	NA

* – Option used

Fig. 8 Task and flight conditions.

Flight task	Flight symbology								
	Hover mode		Cruise mode						
	AH-64 hover		AH-64 FHL FCR	AH-64 FHL CRB	AH-64 CHL PL MCR	AH-64 CHL PL HD	AH-64 CHL PL FCR Wings	AH-64 CHL PL, VV FCR Wings	AH-64 HUD
	Sym screen stab.	World ref VV & AC							
Position recovery	*	*							
Low alt spatial awareness			*	*	*	*			
Contour flight			*				*	*	
Up and away spatial awareness			*	*	*	*			*
Spatial orientation			Symbology not applicable for this task						

*	–	Symbology selected	MCR	=	Moving Compass Rose
VV	=	Velocity Vector	HD	=	Heading Digits
FHL	=	Fixed Horizon Line, standard AH-64	HUD	=	Head-Up-Display
FCR	=	Fixed Compass Rose, standard AH-64	AC	=	Acceleration Cue
CRB	=	Compass Rose Box	Wings	=	Aircraft Reference
PL	=	Pitch Lines	CHL	=	Conformal Horizon Line

Fig. 9 Test matrix.

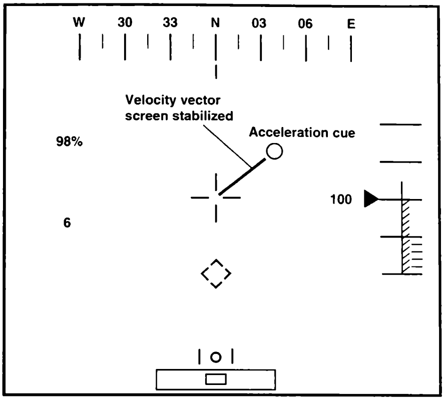


Fig. 10 AH-64 hover flight symbology.

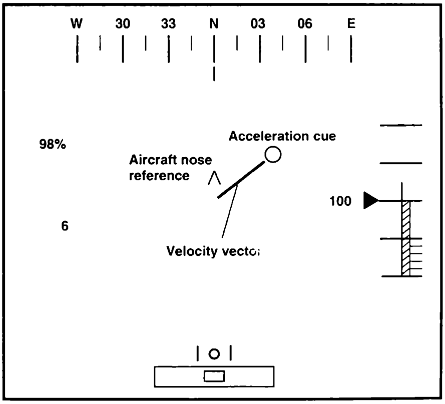


Fig. 11 World referenced AH-64 hover flight symbology.

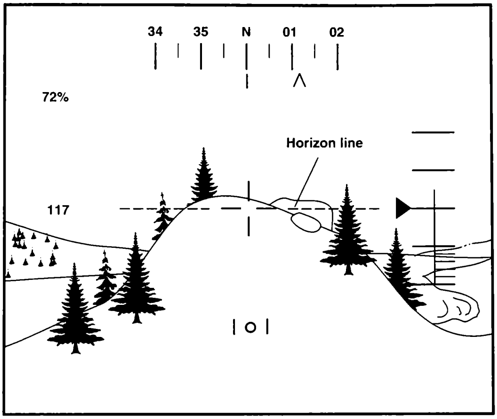


Fig. 12 AH-64 fixed horizon line.

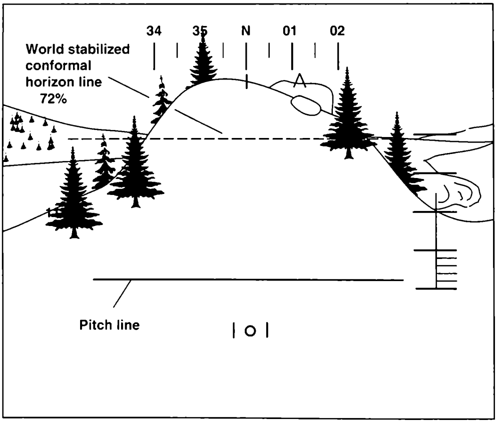


Fig. 13 World conformal horizon line.

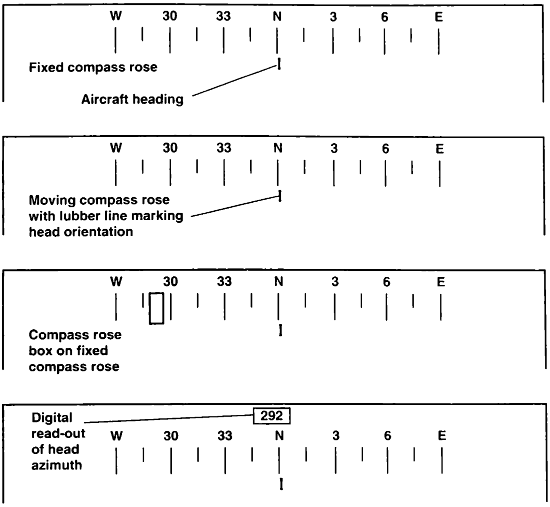

W
30
33
N
3
6
E
Fixed compass rose
Aircraft heading
Moving compass rose with lubber line marking head orientation
Compass rose box on fixed compass rose
Digital read-out of head azimuth
292

Fig. 14 Compass rose head azimuth indications.

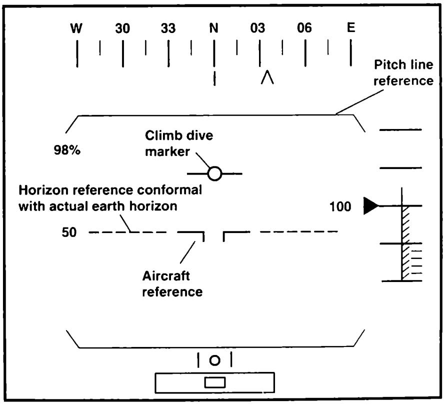


Fig. 15 AH-64A symbology with conformal horizon line, pitch lines, wings, and climb dive marker.

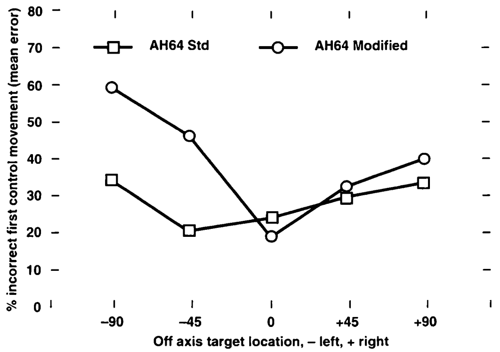


Fig. 16 Position recovery task.

Off-axis spatial estimation from aircraft centerline

Performance measures	– degree left			+ degree right	
	–90	–45	0	+45	+90
Time to correct control movement (seconds)	A 1.3	B .88	B .61	B .84	B .95
Elapsed trial time (seconds)	A 20	B 16	C 15	B/C 15	B 18
Cumulative distance traveled to end of trial (meters)	A 39	B/C 30	C 26	C 28	B 33
Straight line distance from release to end of trial (meters)	A 22	A/B 21	B 20	B 20	B 20
Time until first hover achieved (seconds)	A 8.8	B 7.2	C 5.9	C 6.3	A 8.4
Cumulative distance to first hover (meters)	A 32	B/C 26	C 22	C 23	B 28

Means with the same letter are not significantly different

Fig. 17 Target location/performance measures.

Azimuth and elevation estimation accuracy
Flight symbology set

Mean	STD AH-64 FHL FCR	CHL PL MCR	FHL CRB FCR	CHL PL HD FCR
Azimuth (degrees)	A 5.5	B 1.1	B 1.0	B 0.5
Elevation (degrees)	A 2.2	B 0.6	A 2.4	B 0.5

Means with the same letter are not significantly different

FHL	=	Fixed Horizon Line, standard AH-64	MCR	=	Moving Compass Rose
FCR	=	Fixed Compass Rose, standard AH-64	HD	=	Heading Digits
CRB	=	Compass Rose Box	CHL	=	Conformal Horizon Line
PL	=	Pitch Lines			

Fig. 18 Low altitude elevation and azimuth task.

Azimuth and elevation estimation accuracy
Flight symbology set

Mean	STD AH-64 FHL FCR	CHL PL MCR	FHL CRB FCR	CHL PL HD FCR	HUD
Azimuth (degrees)	B 7.8	C 1.3	C 1.5	C 0.8	A 10.4
Elevation (degrees)	A 1.4	B 0.2	A 1.8	B 0.3	A 1.4

Means with the same letter are not significantly different

FHL	=	Fixed Horizon Line, standard AH-64	MCR	=	Moving Compass Rose
FCR	=	Fixed Compass Rose, standard AH-64	HD	=	Heading Digits
CRB	=	Compass Rose Box	HUD	=	Head-Up-Display
PL	=	Pitch Lines	AC	=	Acceleration Cue
			CHL	=	Conformal Horizon Line

Fig. 19 Up and away spatial awareness estimation results.

Off-axis spatial estimation from aircraft centerline
(–) degrees left (+) degrees right

Mean	–90	–30	0	+30	+90
Error (degrees)	B 4.9	A 10.1	C 1.5	B 6.7	B 5.6

Means with same letter not significantly different

Fig. 20 Spatial estimation error.

AIAA-92-4138-CP

TAKEOFF PERFORMANCE MONITORING SYSTEM DISPLAY OPTIONS

David B. Middleton*
NASA Langley Research Center
Hampton, Virginia

Raghavachari, Srivatsan**
Vigyan, Inc.
Hampton, Virginia

Lee H. Person, Jr.+
NASA Langley Research Center
Hampton, Virginia

Abstract

This paper summarizes the development of displays for an airplane Takeoff Performance Monitoring System (TOPMS) from infancy to its current stage of readiness for commercial application. TOPMS displays provide graphic information concerning the airplane's runway performance and indicate the status of pertinent systems. The TOPMS algorithm computes a nominal performance based on expected (existing) conditions and then compares it to measured performance. Optionally, it also provides "Go/No-Go" advice and continually updates a prediction of where the airplane can be braked to a stop. The TOPMS development experience involved formulating/verifying the algorithm; establishing design criteria and symbology for heads-up and heads-down TOPMS displays; implementing candidate displays in the NASA Langley Transport Systems Research Vehicle (TSRV) B-737 Simulator; arranging for numerous government, airline, and industry pilots to evaluate selected displays on this simulator; conducting TOPMS briefings, discussions, and a workshop with government, airline, and aircraft industry officials; and flight testing a selected set of heads-down TOPMS display configurations on the TSRV B-737 research airplane. Currently, the TOPMS research effort at NASA is being closed out with the evaluation of a baseline display configuration and two alternative configurations on the TSRV Simulator. The display preferences and associated comments of six evaluation pilots are presented and discussed.

Introduction and Background

In recent years, airplane flight safety has improved in most segments of flight, but not during takeoff or abort operations. According to National Transportation Safety Board (NTSB) records, there were over 4,000 takeoff-related accidents between 1983 and 1990, resulting in 1378 fatalities. Among large airliners, 8.7 percent of all accidents occurred during takeoff, and the corresponding rate was 12.5 percent for regional airliners.[1]

Current flight management systems do not comprehensively or effectively monitor airplane performance on the runway. In particular, they do not provide the pilot with timely knowledge of his measured along-track acceleration relative to a nominal acceleration (based on existing conditions and correct execution of the takeoff maneuver). They also do not provide any explicit "Go/No-Go" decision aids. Thus, it was postulated that many serious takeoff-related accidents might be avoided or downgraded to relatively simple low-speed aborts if an appropriate takeoff performance monitoring system (TOPMS) were made available to the flight crew.

* Senior Research Engineer, Member AIAA

** Research Engineer, Member AIAA

+ Senior Research Pilot

TOPMS type research has been conducted[2-5] and discussed for a number of years, but no satisfactory real-time monitoring system has been developed and implemented commercially. The objective of the NASA TOPMS studies has been to develop a TOPMS that would be both useful to and receptive by pilots and also attractive to airplane owners and operators. The TOPMS has been developed at NASA as a software system that could be integrated into flight computer(s) of a modern airplane that already possesses suitable sensors, transducers, databuses, etc.; otherwise this type of equipment would need to be added to support the TOPMS. Evolutionary development of TOPMS technology is summarized in this paper -- from concept to its current stage of readiness for potential commercial application. Accordingly, the paper is structured to provide (1) a brief description of the algorithm, (2) a discussion of TOPMS display features that were preferred or disliked by 41 government and industry evaluation pilots[6,7] and (3) presentation of a recommended "final" display configuration. This final configuration comprises basic status and runway performance information, to which predictive and advisory symbology can be added as options. Simulator evaluation results of an evaluation of these displays (and options) by five NASA research pilots and one airline pilot are also presented.

TOPMS Algorithm

A TOPMS algorithm was formulated[8] to create and manipulate a variety of status, advisory, predictive, and performance information prior to and during the takeoff roll and/or abort. Prior to beginning the takeoff roll, the computer housing this algorithm accepts input values defining existing ambient conditions, runway geometry and pavement conditions (wet, dry, etc.), and aircraft loading. From these inputs and other stored information, including data from the airplane's flight manual, the algorithm determines nominal values for such parameters as engine-pressure-ratio (EPR), critical engine safety speed (V_1), rotation speed (V_R), and takeoff safety speed (V_2). It also computes (predicts) positions on the runway where the airplane will reach V_1 and V_R relative to the selected start point and to a computed ground-roll limit line (GRLL). The GRLL represents the greatest distance down the runway where the airplane can reach V_R and still be able to make an FAA certified takeoff[9] (viz, over a 35'-high barrier at runway threshold with a single engine failure at V_1.)

Prior to brake release, the algorithm computes whether or not the runway is long enough to accommodate the planned takeoff and determines whether or not the flaps are configured correctly. It then computes a nominal acceleration schedule based on existing conditions and correct execution of the takeoff maneuver. During the takeoff roll, it tracks measured acceleration for comparison; then based on this comparison, pertinent runway parameters, constraints, and other programmed criteria, the algorithm continually (1) determines whether the takeoff roll should be continued or rejected (i.e., "aborted") and (2) updates its prediction of the location where the airplane can be stopped using maximum wheel and aerodynamic (spoiler) braking. No credit for reverse thrust is included in the stop-point predictions; however, if both engines appear to be operating satisfactorily, reverse thrust can be used to shorten the stopping distance.

Guidelines and Design Criteria for Displays

The following guidelines and criteria were established for designing, formatting, and implementing the TOPMS displays on the NASA TSRV B-737 fixed base simulator and subsequently on the TSRV itself:

1. The displayed information and format (DIF) should be applicable to a modern 2-engine airplane.
2. The DIF should comprise only status, performance, predictive, and advisory information pertinent to the immediate takeoff-roll and/or abort task.
3. The DIF should provide graphic warnings against beginning a takeoff roll if the assigned runway is too short, if any control surfaces are incorrectly configured, or if the input data set appears to contain invalid elements (e.g., unreasonable values).
4. Duplication of information appearing elsewhere in the cockpit is permissible;

however, the TOPMS DIF as a whole should be kept simple and uncluttered.

5. The meaning of all elements of displayed information should be immediately apparent, and monitoring the display should not appreciably add to the pilot's mental workload.
6. To the extent possible, the DIF should be analogous to and in concert with real-world visual surroundings; use of abstract/digital display information should be minimized.
7. Advisory information should be clear and precise; accordingly
   - a) The following were identified as "No-Go" (abort) situations:
     1. both engines fail,
     2. one engine fails while airspeed is less than V_1, and there is adequate distance left to stop on the runway surface,
     3. predicted position where V_R will be reached is beyond the GRLL, and/or
     4. acceleration error is greater/less than specified acceptability values.
   - b) The following were identified as "Go" (i.e., "do not abort") situations:
     1. airspeed greater than V_1, and/or
     2. predicted stop point beyond end of runway.
8. When using a very long runway, the pilot should be offered the choice of "Go" or "No-Go" if one engine fails at or near V_1 (current FAA regulations[9] require takeoff if airspeed is greater than V_1).

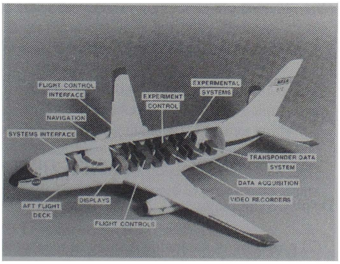


Figure 1. Functional view of TSRV B-737

A cutaway functional view of the NASA TSRV B-737 airplane is pictured in figure 1. The TSRV software was implemented on the computer console labeled "Displays".

Display Evaluation Studies

Takeoff performance and selected systems status, predictive, and advisory information were conveyed to the pilots on Heads-Down Displays (HDD) in the TSRV Simulator cockpit and a Heads-Up Display (HUD) mixed into an "out-the-window" video runway scene. Both pilots had HDD's, but only the pilot making the takeoff had the HUD.

Initial HDD Experience

Symbology for the TOPMS HDD is defined on figure 2. The HDD's in the initial simulation study[6] contained everything on this sketch except the EPR bars; they were added for the reference 7 study. When the Situation Advisory Flag (SAF) is red (indicated by diagonal stripes in the figure), an abort is being advised. In the figure 2 case, the airplane is advancing at 97 knots airspeed toward two triangular symbols. The apex of the unshaded triangle marks where the airplane's speed was initially predicted to reach V_R. The (updated) dark triangle has moved forward from the unshaded one to mark the currently expected position for reaching V_R. Movement of the dark triangle relative to the (fixed) unshaded one provides the pilot with fundamental performance information; in the figure 2 case, it indicates that less than nominal acceleration is being achieved. Further scrutiny reveals that the engine flags are green (denoted by horizontal stripes in this sketch) and that the EPR bars are at their target length; therefore the problem does not appear to be thrust related. (Such a determination was not as easy to make during the reference 6 study because the EPR bars were not available). The dark triangle has not crossed the GRLL, so the remaining runway length (see Criterion 7a-3) is still sufficient for takeoff. Consequently, excess drag and/or some combination of undetected errors are suspected. The red SAF indicates that the acceleration deficiency is serious, and that abort is the appropriate control action.

Whenever an abort was executed, the displays (see right side of fig. 2) showed two predicted stop points: one for stopping with maximum braking ("X") and the other (oval shaped) based on continuation of the currently measured level of deceleration (i.e., braking) for the remainder of the run. During abort maneuvers, airspeed is replaced in the speed box by groundspeed.

Using a detailed TOPMS display-rating diagram (containing display criteria and a 1-10 rating scale (where "1"=excellent) and an associate questionnaire, 32 government and industry evaluation pilots, operating as 2-man crews gave the initial HDD's an average rating of "3" (characterized as"satisfactory--good").[6] Their perception was that the TOPMS HDD's were easy to monitor and provided valuable status, performance, and advisory information - timely information not currently available in their cockpits. They recommended continuation of TOPMS research and suggested the following:

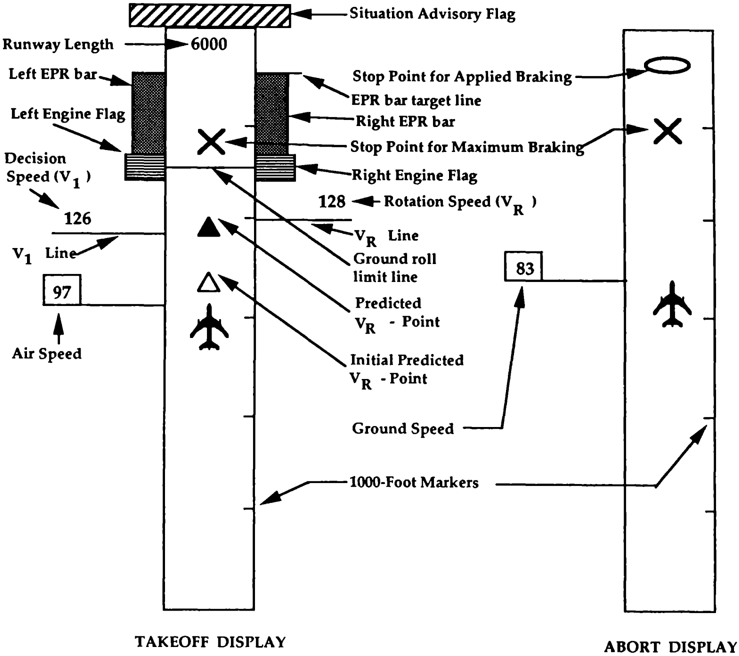


Figure 2. Initial TOPMS Display Symbology

(1) development and implementation of a simplified TOPMS DIF for use in a heads-up display (HUD) mode,

(2) provision of an analog display of measured-EPR relative to an EPR-target value (i.e., the "scheduled" value obtained from the airplane's flight manual), and

(3) provision of along-track-acceleration error as an analog display symbol.

HUD and Modified HDD Experience

A simplified HUD and a modified HDD were implemented on the TSRV Simulator and evaluated by 17 pilots (including 8 from the original group).[7] Figure 3 shows the relatively simple TOPMS HUD superimposed on the runway scene during a normal, no-error takeoff roll - wherein the two triangles are superimposed and the EPR bars were extended to the proper length. Airspeed is shown as a large numeral and the other symbols defined on figure 2 are not included on the HUD.

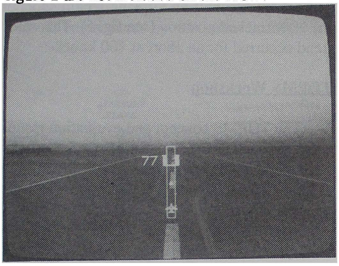


Figure 3. HUD Depicting Normal Takeoff Roll

Modifications to the HDD included addition of the EPR bars and EPR target lines. The HDD was again rated "satisfactory - good", and the HUD was rated "satisfactory - very good". The conclusions from these two experiments were: (1) the TOPMS displays provided appropriate and adequate information to assist the pilot in his Go/No-Go decision and (2) as his attention is normally directed out the forward window, it is preferrable to an HDD.

Many additional recommendations were made and considered. The following were selected and implemented for further study

(1) Delete the engine status flags (see fig. 2); for monitoring purposes, the EPR bars and EPR-target marks were to provide adequate engine performance information.

(2) Display no SAF when a takeoff is proceeding satisfactorily; only display the green Go-SAF in cases where continued takeoff becomes the only viable option (e,g., when the predicted stop point is beyond the end of the runway).

(3) Make the "Go" and "No-Go" SAF's different size and/or shape so they can be quickly distinguished in poor lighting conditions without dependency on color.

(4) Omit the amber Go/No-Go SAF resulting from Criterion 7c. (Several pilots commented that their mind becomes set to "Go" when airspeed nears V_1 , and additional advice at this critical stage of the takeoff is distracting.)

TOPMS Implementation on TSRV

After modification of the HDD in accordance with the above recommendations, it was implemented in the research (aft) flight deck of the TSRV B-737 research airplane (see fig. 1) where it was displayed on the CRT-type display device shown in figure 3. In this airplane, there are no provisions for installing either HDD's or HUD's in the normal forward (safety) flight deck nor provisions for braking or showing HUD's or out-the-window runway scenes in the research flight deck. HDD experiments only were conducted during the flight tests. An example situation during takeoff is shown in figure 4, wherein the engine on the right side has failed at approximately 104 knots and the

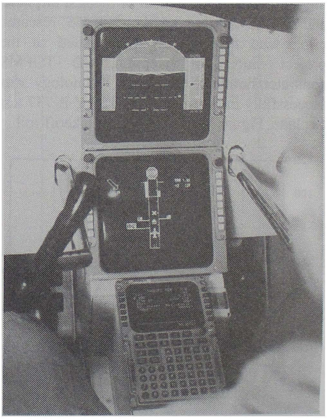


Figure 4. TOPMS HDD on TSRV B-737 Navigation Display screen

right EPR bar has diminished in length and turned red. A red SAF (STOP sign) has appeared, advising the pilot that the best control option is to abort the takeoff and the predicted maximum-braking (wheels and spoilers) stop point ("X") is shown to be about half-way down the runway.

Flight tests were made at five different airfields under a large range of ambient conditions and several types of dry-runway surfaces, including asphalt, smooth concrete, and concrete with longitudinal and lateral grooves (up to 1/4" deep and wide). No wet or icy surfaces were encountered.

The accuracy of the distance predictions and computation of the instantaneous location of the airplane on the runway were checked during several runs at the NASA Wallops Flight Facility at Chinquoteague, Virginia using a highly accurate Laser Oblique Radar Tracker.

The TOPMS HDD performed quite well in the real-world environment of the test airplane, and in the opinion of the TOPMS research pilot, who has actively participated in the project since its inception, the TOPMS computer/simulator-developed technology was successfully transferred to the TSRV B-737 test airplane. Figure 5 shows a typical takeoff roll to 80 knots, where the run was aborted and the airplane was braked to a stop using maximum braking. The laser tracker data showed that the airplane stopped about 2400' down the runway.

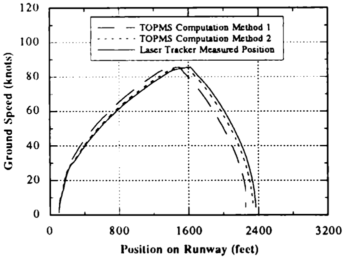


Figure 5. Takeoff/abort travel distance as determined by 3 methods

The TOPMS algorithm, after filtering and double-integrating measured acceleration (Method 1), computed the stop point to be about 1.5 airplane lengths less than that measured by the laser tracker. This amount of under-prediction may be marginally acceptable, but the apparent error (due to sensor and/or TOPMS filtering/computing) was not in the conservative direction. In post flight analysis, it was determined that if the independently measured/filtered groundspeed signal (available on this airplane) had been single-integrated (Method 2), the airplane's position would have been about one-half an airplane length short of the laser tracked position (see fig. 5). This same trend occurred for an abort at 100 knots.

TOPMS Workshop

The TOPMS concept and evaluation results were presented at a national workshop for airline management/engineering, airline-pilot organizations, airplane and avionics systems manufacturers, the Federal Aviation Administration (FAA), the National Transportation Safety Board (NTSB), and the Flight Safety Foundation. The overall reaction was that the TOPMS was the type of cockpit technology that potentially could aid the pilot is making quicker and better decisions concerning continuation or rejection of a takeoff. However, some of the attendees felt that the displays contained more information than necessary or desirable and might be subject to misuse or misinterpretation. In particular, the advisory flags and the predicted stop-points were identified as potential problem sources.

Subsequently, a number of alternative takeoff performance display formats were investigated. The remainder of this report describes experiments conducted in the TSRV simulator to evaluate three of the most promising configurations.

Display Options: Description and Evaluation

Three selected TOPMS display configurations (i.e.,options) were shown to five research and one airline pilot, who flew and observed each HDD/HUD configuration for a variety of simulated conditions. All six pilots had participated in at least one of the previous TOPMS studies. The display options included:

(1) Option 1: contains basic performance and status indicators, but lacks "Go/No-Go" advisory flags and predicted stop-point information; however, during the abort maneuver (only), it continually provides a predicted stop-point based on measured acceleration (this latter feature is common to all three options).

(2) Option 2: expands Option 1 by including "Go/No-Go" advisory flags and another predicted stop-point -- i.e., one based on double integration of the acceleration that would be produced by maximum braking.

(3) Option 3: expands Option 2 by providing a preliminary or "abort warning" octagonal flag (outline of a STOP sign) when degraded acceleration performance exceeds a preliminary threshold; a full abort advisory flag (STOP sign) appears when the error exceeds a second acceptability threshold.

Symbology for Option 1 during a particular situation is illustrated by the sketch on the left in figure 6. At 97 knots airspeed, measured along-track acceleration has become quite deficient, as depicted by the length of the acceleration-error arrow. This arrow is not protrayed when the error is less than 5 percent; above 5 percent, a white arrow appears until its length exceeds limit 1 (shown as a large tick mark under the acceleration-error arrow); between limit 1 and limit 2 (tick mark superimposed on the GRLL) the arrow is shaded amber; and upon reaching limit 2, the arrow turns red. (The limit-2 tick mark has no relationship to the GRLL; in this sketch, the limit-2 tick mark was scaled and arbritrarily placed on the GRLL to preclude the need for an additional tick mark.) As a result of the large acceleration error, the dark triangle, marking the current prediction of where the airplane will reach V_R, has moved farther down the runway. This is considerably ahead of the stationary unshaded triangle that marks the initial or nominal (no-error) location where V_R should have been reached, indicating that acceleration is and/or has been below nominal. However, both EPR bars extend up to the EPR-target line, indicating that both engines are providing nominal thrust. Consequently, the acceleration deficiency is probably due to some type of excess drag (e.g., aerodynamic and/or

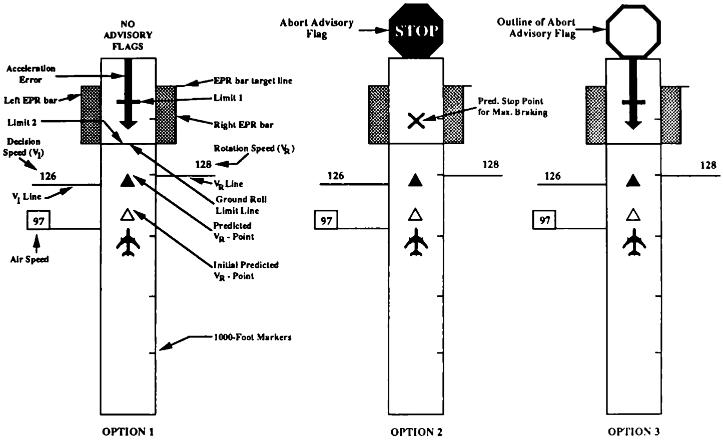


Figure 6. Symbology for TOPMS Display Options

frictional). Even though takeoff remains a viable option as long as the dark triangle does not cross the GRLL, an abort should be given strong consideration -- particularly if the acceleration-error arrow nears limit 2. No situation advisory flags are provided in Option 1, but limit 2 could be considered an abort threshold.

Option 2, shown in the center of figure 6, adds situation advisory flags (SAF's) and a predicted stop-point "X" based on immediate abort and maximum braking. The situation is similar to that for Option 1, except that the acceleration-error has exceeded its programmed acceptability limit and disappeared from view. Simultaneously, an abort SAF (STOP sign) appeared, advising an immediate abort, which can be terminated at the "X" symbol. (If the "X" had appeared beyond the end of the runway, a large green forward-pointing arrow ("GO" advisory flag) would have appeared in place of the STOP sign.)

Option 3, on the right in figure 6, is a variation of Option 2. The situation and the acceleration-error limits are the same as shown in Option 1. However, the display portrays a preliminary abort-advisory flag in the form of an unshaded octagon which appears whenever the acceleration-error arrow reaches limit 1. A full abort SAF (STOP sign) will be portrayed whenever the error arrow reaches limit 2.

An example of both heads-up and heads-down Option 1 displays in the TSRV B-737 Simulator is shown in figure 7. An apparent excess-drag situation is depicted; the situation will be portrayed whenever the error arrow reaches limit 2. The arrow has turned amber, and will turn red if it reaches limit 2. Option 3 displays for the same situation are shown in figure 8 with the primary difference being that a predicted stop-point ("X") and a preliminary abort SAF (outline of an octagon) have appeared. (Whenever the error arrow reaches limit 2, the octagon will convert to the full abort SAF (STOP sign), and the display will then look exactly like the Option 2 display.)

Once an abort has been initiated while using any of the options, all of the symbology related to takeoff is deleted. However, the airplane (position) and stop-point symbol(s) are retained, but, groundspeed replaces airspeed in the box to the left of the airplane symbol. Option 1 contains only a predicted stop point (oval shaped) based on measured acceleration. Options 2 and 3 contain this symbol as well as an "X" symbol, denoting a predicted stop-point based on maximum wheel and spoiler braking.

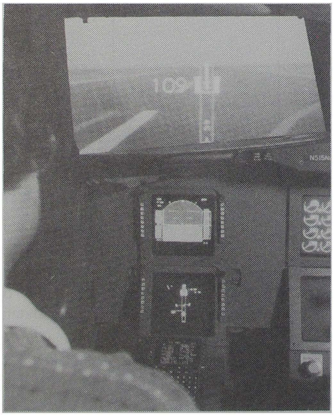


Figure 7. Option 1 display showing marginally acceptable acceleration error

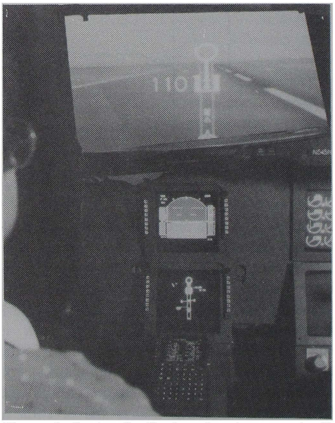


Figure 8. Option 3 display showing marginally acceptable acceleration error

Results and Discussion

The pilots in this study did not perform a rigorous evaluation of the displays as they had in prior TOPMS studies[6,7]; instead, they ranked the three options according to preference and then answered a short questionnaire supporting their choices. Their rankings are presented in the table below; a summary of their comments follows:

Pilot Rankings of Display Options

	OPT 1	OPT 2	OPT 3
Pilot 1	3	2	1
2	1	3	2
3	3	2	1
4	1	2	3
5	3	2	2
6	3	2	1

Pilot 1

* Preferred Option 3 over 2 because of the way in which acceleration-error degradation is depicted:
  - appearance of the octagon (outline of STOP-sign) appropriately warns and mentally prepares him for potential abort situation;
  - such symbology is "less dictatorial" than having red SAF (STOP sign) suddenly appear (Option 2) whenever acceleration error exceeds an unacceptability limit.

* Acceleration-error arrow color changes in Option 1 are less noticeable/obvious than appearance of the octagon in Option 3, and would not prompt him to abort as quickly.

* Preferred having more display information than provided by Option 1; therefore prefers Option 2 over 1;

* Preferred having both stop-point symbols appear in all abort displays.

Pilot 2

* Preferred Option 1 displays, but not strongly; liked certain features of each.

* Preferred Options 1 and 3 over Option 2 during degraded acceleration-performance situations because their displays provide enhanced indications of acceleration degradation.

* Judged Option 1 to provide generally adequate information at low and moderate speeds, but at higher speeds, prefers having abort decisions reinforced by the Go/No-Go SAFs (Options 2 and 3):
  - would like the SAF to remain on the screen longer during abort-initiation maneuver.
  - preferred seeing the maximum-braking stop point ,"X", during high-speed aborts.

* Preferred that stop-points not be shown during normal takeoffs.

Pilot 3

* Considered all three display options satisfactory and desirable; with proper training, all can be used comfortably and successfully.

* Preferred displays providing the most information

* Option 1 judged to provide generally adequate information. However, for problems at high speeds, lack of SAF might cause loss of valuable time "visually troubleshooting" while the airplane was rapidly running out of potential braking space.

Pilot 4

* All options judged to provide adequate and timely information; the associated displays were simple, intuitive, and easy to monitor.

* Preferred Option 1 because of its simplicity; can recognize potential abort situations with it about as quickly as with options containing abort SAF's.

* Recommended that the "must-Go" SAF (large green forward-pointing arrow) be added to Option 1 to quickly alert the pilot whenever continued takeoff becomes the most viable control choice.

Pilot 5

* Symbology for all three options judged to be satisfactory and operationally acceptable.

* Preferred Options 2 and 3 at high speeds because the SAF's provide a definite indication of serious problems.

* Judged abort conditions at low speeds to be obvious; therefore, prefers Option 1.

Pilot 6 (airline pilot)

* Judged that very little training would be required to fly all three options equally well.

* Preferred having as much information as possible available in case of need.

- * Judged the predicted maximum braking stop point ("X") during abort to be useful in determining amount of pressure on brake pedals required to produce full braking.
- * Judged the amber acceleration-error arrow (Option 1) and the octagon outline (Option 3) to not remain on the screeen long enough; recommended using a 10 percent differential between limit 1 and limit 2, instead of the 5 percent used in this study.

Pilot evaluation results are summarized and discussed by option in the following paragraphs.

Option 1 was favored by two of the six pilots. In their opinion, this option provided all necessary information: runway locations where V_1 and V_R would be reached; measured acceleration-performance relative to predicted performance; critical engine status and performance information; and a pictorial indication of minimum field length required for takeoff. Lack of explicit "Go/No-Go" advice or an indication of where the airplane could be stopped in case of abort were not considered display deficiencies by these two pilots. While the other four pilots did not dislike this option, it was ranked third because the least amount of information was provided.

Option 2 provided essentially the same information as Option 1 plus "Go/No-Go" advisory cues and predicted maximum-braking stop-point information. Because it provided this additional information, five of the six evaluation pilots rated it their second preference. It was ranked third by the other pilot because it lacked the acceleration-error enhanced gradation symbology provided by Options 1 and 3.

Option 3, very similar to Option 2, adds an advance-warning abort SAF to indicate the existence of acceptable, but less than desirable acceleration. This option was preferred by four of the six pilots, primarily because of this feature which allowed extra time to mentally prepare for a potential abort.

Even though the pilot evaluators in this study only ranked the TOPMS display options, it was clear from the verbal and written comments that the displays were found to be readily comprehendible, easy-to-learn and easy-to-use cockpit technology. Any of the three display options investigated were judged to be satisfactory from a control-decision standpoint and, with suitable training, could be used comfortably and confidently. Although each pilot had certain display preferences, the consensus judgment was that providing any appropriate set of performance and status information during takeoffs/aborts is far better than current-day cockpit technology. Also, most of the pilots in the current study judged that, while providing predictive and advisory information was appropriate and useful, the displays had considerable merit with or without it. Similar comments were made at the end of each of the prior TOPMS simulation studies.[6,7]

Concluding Remarks

In prior NASA TOPMS studies, many display elements were viewed, "flown", and evaluated by a significant number (41) of Air Force, airline, and research pilots. They recommended elimination of symbology considered to be of limited value and addition of other useful features. Many of these features have been implemented and the displays have evolved to the three options investigated in the current study.

This investigation concludes the TOPMS formal development and evaluation program at NASA Langley. The algorithm has been judged to provide valuable and appropriate takeoff/abort performance and advisory information. The TOPMS heads-up and heads-down information displays have evolved into an expandable "final" configuration set comprised of a baseline array of elemental information, with options to include certain advisory and predictive features. Three such options were investigated in this study. All were judged acceptable by six evaluation pilots. However, there was not a consensus on the preferred option. In particular, four pilots preferred the option (Option 3) that provided the most information -- i.e., all of the elemental, predictive, and advisory information. The other two pilots selected the Option 1 displays, which provided only enhanced elemental information. Option 2, which provided both elemental and advisory information was selected as a strong second choice by five of the six pilots.

As a consequence of this study and earlier concerns expressed by the aircraft community, it seems prudent not to recommend a single "final TOPMS configuration." Instead, all three study configurations are presented as viable candidates. Thus, potential users may select the features that best suit their needs and operations philosophy. Their "selected configuration" may be one of the configurations investigated in this or prior TOPMS studies; or it may comprise some other combination of the above discussed features; or it may only contain part of them. In the opinion of the evaluation pilots, there would be no problem learning to use any of the options and each could be used confidently and successfully. Each pilot emphasized that use of any one of the options was far better than having no TOPMS-type information in the cockpit. The TOPMS algorithm is designed to service any of these and many other options. Consequently, additional display features, such as advisory or predictive information, could be readily incorporated.

References

1. Jacque, Peter: "NASA Offers Airline Pilots Aid". Richmond Times Dispatch. Nov. 12, 1989, Pg. C-8.

2. Fusca, James A.: "Takeoff Monitor Computes Runway Roll". Aviation Week; Jan. 13, 1958. p. 99-105

3. Aron, Ioan; and Tomescu, Ion: "Takeoff Director System". United States Patent No. 4,251,868. February 7, 1981. Performance Margin Indicator". European Patent Applic. No 85200977.8, Publication. No. 0 166 487 A2, June 19, 1985.

4. Small, John T, Jr.: "Feasibility of Using Longitudinal Acceleration (N_X) for Monitoring Takeoff and Stopping Performance From the Cockpit". The 1983 Report to the Aerospace Profession - 27th Symposium Proceedings, vol. 18, no. 4, Society of Experimental Test Pilots, c.1983, pp 143-145. (A84-16157 05-05).

5. Boeing Company (Patrick J. Cleary, Lloyd S. Kellam, and Richard L. Horn): "Aircraft Performance Margin Indicator". European Patent Applic. No. 85200977.8, Publication No. 0 166 487 A2. June 19, 1985.

6. Middleton, David B.; and Srivatsan Raghavachari: Evaluation of a Takeoff Performance Monitoring System Display. Journal of Guidance, Control and Dynamics, Volume 12, No. 5. Sept.-Oct. 1989.

7. Middleton, David B. ; Srivatsan, Raghavachari; and Person, Lee H., Jr.; "Simulator Evaluation of Takeoff Performance Monitoring System Displays"; Paper No. 88-4611-CP, Presented at AIAA Simulation Technologies Conference, Atlanta Georgia, Sept. 9-11, 1988.

8. Srivatsan, R.; Downing, David R.; and Bryant, Wayne H.: Development of a Takeoff Performance Monitoring System. NASA TM 89001. 1986

9. Airworthiness Standards: Transport Category Airplanes. FAR Part 25.55, Federal Aviation Administration. June 1955.

AIAA-92-4140-CP

USE OF HIGH-FIDELITY SIMULATION IN THE DEVELOPMENT OF AN F/A-18 ACTIVE GROUND COLLISION AVOIDANCE SYSTEM

Timothy R. Fitzgerald*
Michael T. Brunner*
Strike Aircraft Test Directorate, Code SA103
Naval Air Warfare Center Aircraft Division
Patuxent River, Maryland 20670

Abstract

An active Ground Collision Avoidance System (GCAS) has been developed for the F/A-18 using the Naval Air Warfare Center's F/A-18 simulation. The simulation has been used for the development of all three components of GCAS: (1) the algorithms used to determine the recovery initiation altitude; (2) the additional flight control laws (FCL's) necessary to perform the recovery maneuver; and (3) the visual and audio cues used to provide recovery status information to the pilot. The use of a simulation has allowed the rapid development of a viable F/A-18 GCAS that incorporates technology from the F/A-18 Integrated Fire and Flight Control (IFFC) simulation and the Advanced Fighter Technology Integration (AFTI) F-16 program. Complete system development and preliminary evaluations were performed using the simulation. This increased overall project safety while decreasing development and potential flight test costs significantly.

Introduction

In the United States Navy alone, as many as nine aircraft were lost as a result of controlled flight into terrain (CFIT) during 1988. The pilots became distracted, saturated by their workload, incapacitated, or simply "flew into the ground", resulting in the loss of lives and millions of dollars of hardware. Clearly, with the advanced control systems of today's aircraft, these losses are not only tragic but unnecessary.

Flight tests conducted with the AFTI F-16[1] have shown that an increase in safety could be realized with the use of an active GCAS rather than the existing passive systems. Developed mainly as a fail-safe against pilot loss of situational awareness and g-induced loss of consciousness (GLOC), the Navy saw the potential application toward its own tactical aircraft, in particular the F/A-18, which has more than twice the incidence of GLOC per 10,000 flight hours (12.9) than any other aircraft in the Navy inventory[2]. With its digital flight control system, the F/A-18 is extremely adaptable to new control technologies, and the decision to fund a proof-of-concept active F/A-18 GCAS was made during fiscal year 1990. Furthermore, development of this active GCAS and its subsequent evaluations against the F/A-18's current low altitude warning system would be done exclusively using a simulator.

The Manned Flight Simulator (MFS) facility located at the Naval Air Warfare Center Aircraft Division (NAWC/AD) has developed a high-fidelity, non-linear, real-time F/A-18 engineering simulation. The simulation has been used in support of numerous Navy flight test projects, investigations of various Navy fleet incidents, and the National Aeronautics and Space Administration High Angle-of-attack Research Vehicle program. The facility includes an actual F/A-18A cockpit; a 40 foot (12 meter), 360° field-of-view dome; a six-degree-of-freedom motion platform; and a CompuScene IV image generation system.

This paper shall address the advantages of using simulation to develop and evaluate an active GCAS, and present results, recommendations, and lessons learned.

GCAS Design Philosophy

An overview of the GCAS implementation is shown in Figure 1. The additional FCL's are external to the flight control computers (FCC's), and the GCAS recovery commands are summed into the longitudinal and lateral axis stick command paths of the existing FCL's. The FCC would use the summed GCAS and pilot commands in the same way it currently uses only the pilot command. The GCAS control loop is closed by feedback of measured aircraft responses provided by current aircraft sensors; no additional instrumentation would therefore be necessary. GCAS is active only when the aircraft altitude is at or descending below the computed recovery initiation altitude; it remains active until the aircraft achieves a positive 5° flight path angle. This angle was chosen to provide a positive rate of climb after pull-up at nominal pitch attitudes throughout the flight envelope while maintaining a sufficient energy state.

An auto-recovery system must perform two distinct tasks: (1) decide when a recovery maneuver must be initiated in order to pull out at or above the pilot-designated floor altitude; and (2) supply control system commands to perform the maneuver. The design philosophies for each of these tasks, along with the recovery system cuing, follow.

Recovery Altitude Calculations

The algorithms of the Straight-Forward Auto-Recovery System (SFARS)[3] were found to be best suited for the GCAS recovery altitude computations. The simulation was used to determine the aircraft-specific gain schedules used in these algorithms. The SFARS algorithm determines the altitude that will be lost during a given recovery maneuver (Δz). This is then added to the pre-selected floor altitude to obtain the recovery initiation altitude. The original algorithm used six independent components that compensated for dive angle, bank angle, g-onset, sensor lag, excess (i.e. non-idle) power, and roll rate at recovery initiation.

Initially, two of the six components were neglected from the system build-up: sensor lag and roll rate at recovery initiation. Potentially, the most crucial sensor lag would be that of the mean sea level (MSL) altitude sensor; the above

* Aerospace Engineer; Member AIAA

ground level (AGL) sensor is not used at this time because of its limited coverage in some aircraft attitudes. Based on AFTI flight test results[3], the MSL sensor was found to have little or no time lag so this compensation term could be neglected. The roll rate compensation term was neglected since this system is designed for recovery from CFIT situations and these situations typically have low roll rates. In the future, if this system is expanded for use in other situations, the roll rate term may become important. Later, the excess power compensation was also excluded in favor of using the F/A-18's Automatic Throttle Control (ATC) system, as will be discussed later.

The derivations of the original SFARS equations can be found in References 1 and 3. The final form of the F/A-18 derivative system is shown, in block diagram form, by Figure 2.

The gain schedules for dive angle and g-onset compensation were determined first, followed by the bank angle and excess power compensation terms. The entire system was optimized for the fighter escort (FE) configuration, an air-to-air store loading.

The dive angle compensation term is computed using the following:

$$\Delta z_1 = \frac{V_{true}^{\,2}\,(1 - \cos\gamma)}{K_1\, g}$$

where V_{true} is true airspeed (ft/sec), γ is the flight path angle (radians), and g is the acceleration due to gravity (ft/sec^2). K_1 was initially defined as a function of calibrated airspeed only. The gain schedule was developed using the simulation to gather recovery data at various airspeeds during a 30° dive. The schedule was optimized so that recoveries at all airspeeds fell within the acceptable recovery window of 200 feet (61 meters) above the floor altitude. After subsequent problems with steeper dive angle recoveries penetrating the floor altitude, the schedule was modified to include flight path angle in the schedule's functionality, as shown by Figure 3.

The bank angle compensation term is computed from the following equation:

$$\Delta z_2 = \frac{|\phi|}{K_2}\,|V_D| + (|\phi| - 45)\,K_\phi$$

where ϕ is the bank angle (deg), V_D is the inertial frame downward velocity (ft/sec), and K_2 is essentially the roll time constant which is scheduled with calibrated airspeed. This schedule is shown in Figure 4 and was also optimized using the simulation to gather data at various speeds and 30° of bank angle in concert with the previously optimized dive angle compensation. The K_ϕ portion of the Δz_2 equation was added after it was determined that the commanded roll rate is not only a function of airspeed, but also the bank angle of the aircraft. The GCAS control laws will command different roll rates depending on the bank angle. Using the excess power compensation term of the original SFARS algorithm as a guide[3], this empirical fit was developed with an intercept of $\phi = 45°$ and a slope (K_ϕ) of 1.3 ft/deg (0.4 m/deg). The K_ϕ portion of this term is retained only if it is positive.

The g-onset compensation term is determined by the amount of time it takes to reach a desired load factor. The equation for the altitude lost during this process is simply:

$$\Delta z_3 = K_3\,|V_D|$$

The value of the K_3 gain is the aircraft's g-onset time constant. The first value used, 1.1 seconds, was that of the AFTI F-16, found in Reference 3. This value turned out to be close enough to the F/A-18's actual time constant of 1.2 seconds that no changes were warranted.

The excess power compensation term was initially included to handle non-idle power situations. Its equation was:

$$\Delta z_5 = \frac{V_{cal}\,a_z - K_{P_s}}{K_5}$$

where $K_5 = 11.85$ scaling coefficient
$K_{P_s} = 9{,}300$ knots-ft/sec^2 (2,834 knots-m/sec^2)

V_{cal} is calibrated airspeed (knots). The value of K_{P_s} is defined as the flight condition where specific excess energy (p_s) is zero at a normal acceleration (a_z) of 128.8 ft/sec^2 (39.24 m/sec^2) or 4 g's, and was obtained from F/A-18 maneuvering diagrams. While this compensation term worked adequately for the more shallow dive angles ($\gamma \leq 30°$), it was found to be less accurate at compensating for the elevated energy levels and accelerations associated with steeper dives. Problems with this term were confirmed through conversations with the author of Reference 1. It was suggested that aircraft equipped with an automatic throttle system may be better served by tying into it rather than compensating for altitude loss in the recovery algorithms. As a result of this suggestion, the F/A-18 GCAS was modified to engage the aircraft's ATC five seconds prior to pull-up in order to allow the engines sufficient time to spool down to idle. Upon selection by GCAS, the ATC drives the throttles to flight idle for the duration of the recovery maneuver. After recovery is completed or if a manual disengagement is commanded, the ATC restores the throttles to their original position. This approach required only minor changes to the ATC control laws to allow engagements outside of the normal authority of the system when GCAS is active.

Flight Control Law Modifications

The additional FCL's necessary for the aircraft to perform the auto-recovery maneuver are shown in detail by Figure 5. These FCL's descended from the early General Electric FIREFLY routines via an F/A-18 IFFC simulation developed by NAWC/AD. Gain schedules for the longitudinal axis FCL's were optimized through the course of several simulation runs at a constant dive angle and various airspeeds to obtain acceptable aircraft responses across the flight envelope to GCAS commands. Both longitudinal and lateral axis systems are simple proportional command systems. Open-loop analysis of this system yielded no safety of flight or stability concerns, however, a closed-loop analysis should be completed before any flight testing begins.

Pilot-Vehicle Interfaces

Pilot-vehicle interfaces with GCAS are kept as simple as possible. The desired floor altitude is pre-selected prior to the simulation run and entered into the system. During real-time runs, the pilot is required to enable the system via a cockpit switch selection. While active, the system is disabled whenever the landing gear is extended, the flight control system (FCS) enters spin recovery mode, or a momentary selection of the control column paddle switch is made by the pilot. The system automatically re-activates upon landing gear retraction or resumption of normal FCS operation in conjunction with a positive climb attitude above the designated floor altitude.

Recovery System Cues to the Pilot

The major recovery system visual cue consists of a pair of chevrons presented on the head-up display (HUD) approximately five seconds prior to initiation of the automatic recovery. As the aircraft continues its descent, the chevrons draw together to form a break "X" at the pull-up altitude (Figure 6). This symbology, chosen because of its simplicity and intrinsic meaning to a tactical pilot, is identical to that used in the AFTI F-16 GCAS flight test aircraft[1]. If the pilot intervenes before the pull-up point by moving the stick in such a way as to delay the onset of the pull-up point (e.g. decreasing the dive angle), the chevrons will begin to part again, indicating that the actual pull-up point is being revised by GCAS. Also, the rate at which the chevrons come together indicates how quickly the aircraft is approaching the pull-up point. The HUD symbology (including the chevrons) was programmed onto a graphics processor and projected ahead of the pilot. The normal HUD "ATC" cue is also provided whenever GCAS engages the automatic throttle control system during a recovery.

An audio voice alert ("ALTITUDE...ALTITUDE"), identical to that already installed in the F/A-18, is provided to the pilot whenever the aircraft penetrates the floor altitude.

Aim and Methodology of Piloted Evaluations

Although the original tasking was to compare the active system against the F/A-18's current low altitude warning system, it was decided that such a comparison would be moot. The current system in the F/A-18 provides only a voice alert *at* the pre-selected floor altitude, ensuring floor penetrations in every case. Thus, piloted evaluations of GCAS were approached from slightly different aspects: (1) to evaluate the automatic recovery system performance, from a qualitative as well as quantitative standpoint; and (2) to qualitatively evaluate manual recoveries performed by the pilots using the cues provided by GCAS. This information could be used to assess the value of GCAS in a passive mode, as well as to determine the most effective cue(s) to use with a passive GCAS.

Targeted test points for evaluation of the active GCAS comprised a matrix of three airspeeds (300, 375, and 450 KCAS), three dive angles (30°, 45°, and 60°), three power settings (idle, military, and maximum afterburner), and five bank angles (0°, 30°, 45°, 90°, and 180°). Table I shows the nine points targeted during the passive GCAS evaluations.

V_{target}	γ	ϕ
KCAS	deg	deg
300	30	30
375	30	45
450	30	0
300	45	0
375	45	45
450	45	180
300	60	30
375	60	0
450	60	45

Table I

Two different store loadings, both in the up/auto (i.e. cruise) configuration, were tested: (1) FE at 36,124 lb (16,385 kg) and 22.1% c.g. as tested; and (2) interdiction (INT), an air-to-ground loading at 46,284 lb (20,994 kg) and 20.93% c.g. as tested. Figure 7 illustrates both of these loadings. The INT loading was tested to identify any important configuration-dependent parameters that may exist, which will be useful in the event separate gains based on store loading must be optimized. Each configuration was flown by at least two different test pilots, with a total of four pilots participating in the evaluations.

During the automatic recovery phase, the pilots were asked to set up the required test points, note the critical parameters (i.e. airspeed at pull-up, minimum altitude during recovery, and peak load factor), and provide qualitative assessments of the system performance.

For the evaluation of the manual recoveries, the pilots were not briefed on which points they would see. Additionally, the simulation's visual system was set up to present a cloud base at 3,500 feet (1,067 meters) MSL such that the pilot would have no visual scene references above that cloud level. Together, these helped to induce a loss of situational awareness to the pilot that would not have been otherwise present had he flown the aircraft into the dive/bank angle condition himself. The pilot was required to look away from cockpit instrumentation while the simulation was set up above the cloud base already in the dive/bank condition desired for the test point. After the simulation run had begun, the pilot was then required to make an assessment of his aircraft's attitude and determine if any immediate action was required. If not, the dive would be allowed to continue until pull-up cues were provided by GCAS, at which point, the pilot would be required to perform the recovery maneuver. For the manual recoveries, the HUD chevrons were identical to those used during the automatic recovery sessions, but the voice alert was provided at the pull-up point rather than at floor altitude penetration. Two manual recovery cuing schemes were evaluated: (1) the voice alert cue only; and (2) both the break "X" and voice alert cues.

Discussion of Results

Figures 8 through 10 present results from the piloted evaluations of the active GCAS in the FE configuration. Figures 11 through 13 contain the results for the INT configuration.

Generally, the FE results were within the design tolerance of 200 feet (61 meters) above the floor altitude. At the lowest airspeed tested (300 KCAS), the system had no penetrations and only one point was out of tolerance. There was a marked increase in the scatter at higher airspeeds and steeper dive angles. It is likely this is due to the many acceleration profiles that can lead into and out of the pull-up point. The calculation of the dive angle compensation term, Δz_1, assumes a constant velocity and therefore a circular trajectory throughout the recovery. Different acceleration profiles will cause scatter in the average velocities of the recovery maneuvers and a corresponding scatter in the recovery altitudes. At higher airspeeds and dive angles, there is a larger range of acceleration profiles, and therefore, wider altitude scatter.

Overall, the INT cases showed only a shift to a more conservative recovery with increasing dive angle, but there were a significant number of floor penetrations at the lowest airspeed tested (300 KCAS). Associated with each recovery at this condition was an angle of attack of 20° to 25° and a normal acceleration of under 4 g's, indicating near stall conditions for that gross weight. This also caused an uncomfortably high post-recovery nose attitude ($25° \leq \theta \leq 30°$), since the system's aim is to bring the aircraft's flight path to a positive 5° attitude. Additionally, the peak angle of attack values achieved during these recoveries exceeded the Naval Air Training and Operating Procedures Standardization (NATOPS) angle of attack limit of 20° for this store configuration. INT results improved significantly at higher airspeeds without exceeding NATOPS maneuvering limits; however, there were still several floor penetrations at the inverted bank angles. This is possibly due to roll rate limiting imposed by the FCL's with this store loading.

Pilot reactions to the system overall were positive. The majority felt comfortable enough with the system to trust it in situations of GLOC and spatial disorientation, as well as operationally in the navigation mode, air-to-air arena, and some limited air-to-ground instances (e.g. training).

A cursory look at two asymmetric store loadings and their effect on active GCAS recoveries was done at the end of the evaluations. The smaller lateral asymmetry, 3,800 ft-lb (5,152 N-m), was achieved by removing one of the wingtip AIM-9 missiles from the FE loading; removal of two of the MK83 bombs from one of the outboard wing stations of the INT loading provided the larger lateral asymmetry of 22,000 ft-lb (29,828 N-m). GCAS was able to effectively recover the aircraft to a wings level climb condition with the smaller lateral asymmetry, however, with the larger one, the system could not achieve a complete recovery. The nose attitude was brought up to level, but the large asymmetry continued to drop the heavy wing during the recovery maneuver, resulting in a "porpoising" motion as one recovery after another was attempted by GCAS.

Use of the F/A-18's ATC for handling elevated power conditions gave far fewer floor penetrations than did use of the original SFARS power compensation term. None of the pilots found use of the system objectionable, however, several commented on the timing and amount of automated power application after recovery. The unanimous suggestion on power application was that the system should select military as the nose passes through the horizon during the recovery. One pilot pointed out that the current ATC system may be inadequate for GCAS use due to system reliability issues or the simple fact that pilot selected throttle friction may prevent the system from engaging, as the throttles must move during ATC operation.

The HUD chevron mechanization was immediately accepted by the pilots. All agreed that the chevrons provided several necessary cues on pending system engagement that could be absorbed through their peripheral view. For instance, the rate of closure provided an easily perceptible cue on the sense of urgency of the situation. Slowing of the closure rate or chevron separation provided useful feedback on if and how well the pilot was affecting the situation when he chose to intervene.

Aural cuing as it applied to both active and passive recoveries was viewed as necessary by the pilots. For the active recovery cases, the pilots preferred to have as many cues as possible to notify them that the system was taking control. During the manual recoveries, many pointed out that often times, pilots are not concentrating on the HUD, but are scanning about their aircraft. In such instances, the aural cue serves to notify them of something requiring action and brings their attention back into the cockpit. All agreed that the aural cue should not be used alone, but in conjunction with other cues (e.g. the HUD chevrons).

Due to the limitations of current simulation technology, no direct comparisons can be made between the active and passive GCAS results. While considerations to visual and tactile cues need not be made with respect to the active GCAS, pilots rely heavily upon these cues to perform maneuvers such as a constant g pull-up to wings level.

Overall, there is no life-threatening condition in a simulator, and the pilots realize this, at least on a subconscious level. Poor resolution and lack of depth perception inherent in domed visual systems makes it difficult for the pilot to evaluate the aircraft's situation and the necessity for action in an unknown dive condition. Also, once a recovery maneuver is in progress, the lack of acceleration cues makes a realistic manual pull-up difficult in a simulator. In general, however, recoveries made by GCAS were more consistently above the floor altitude than were the manual recoveries. It was also observed that the pilots often, either intentionally or inadvertently, pulled more than the system-targeted 4 to 5 g's during the manual FE recoveries. Many times, this had the effect of compensating for the lag associated with human response to provide recoveries above the floor altitude, some even quite conservative. These normal acceleration overshoots occurred more when the only recovery system cue provided was the voice alert. This was not the case with INT manual recoveries due to the fact that both the active GCAS and the pilots tended to load the aircraft up to the normal acceleration limit enforced by the FCC's. In regards to the issue of cuing for manual recoveries, all the pilots agreed that the chevron and voice alert combination was superior to the voice alert only. Results generally supported this, as recoveries were typically better when the chevrons were used as compared to the same conditions using audio only, although there was a large amount of scatter.

Conclusions

The usefulness of simulation has been demonstrated in the development of an active GCAS for the F/A-18. Control system gains as well as the recovery system itself were developed in a timely manner at minimal costs.

The shortcomings of a fixed-based simulator manifested themselves in the testing of the passive GCAS, where the results were clouded by the fact that the visual scene in the dome was not sharply defined and provided no depth perception, and there was no tactile cuing.

In its current form, the active GCAS has demonstrated the potential for reducing aircraft losses due to cases of GLOC or spatial disorientation. Nuisance warnings and nuisance pull-ups have been kept to a minimum. In the operational air-to-air and basic navigational modes, the system is adequate as is, but can be refined further. For air-to-ground, GCAS is currently adequate only for training use. Operationally, it is currently inadequate for the air-to-ground mission, but can be made acceptable with work.

Active GCAS appears to perform a given recovery maneuver more efficiently than a pilot could, with fewer floor penetrations. Passive GCAS is inferior to active GCAS, particularly in instances of GLOC or spatial disorientation, but appears to have its place in other applications. In either case, GCAS is more effective at preventing floor altitude penetrations than the low altitude warning system currently in place on production F/A-18's.

The more cues provided to the pilots about impending trouble and/or active GCAS engagement, the better. The HUD chevron/break "X" mechanization, as implemented for this study, is highly desirable. Voice alerts are also desirable but should only be used to complement the chevrons.

If available and reliable, an automatic throttle system used in conjunction with an active GCAS provides a simpler and more effective method of compensating for elevated power conditions during automatic recoveries.

AGL sensors must be used in the long-term for GCAS to have value over all terrain features with as little pilot intervention as possible.

A GCAS system, whether active or passive, is necessary for military tactical aircraft. The SFARS algorithms developed by the United States Air Force have given the United States Department of Defense a viable and inexpensive system, that due to its generic nature, is easily adaptable for use in all its tactical aircraft. The simulation work presented in this paper has provided the United States Navy an SFARS-derivative system that appears to work on the F/A-18.

Recommendations

Continue simulation work in order to refine the role GCAS will play in the air-to-air and air-to-ground arenas. Investigations into optimum default settings of floor altitudes for each and the merit of passive GCAS in the air-to-ground mission can be made. The take-off and powered approach phases of flight must also be considered with respect to GCAS. Passive GCAS may initially be the best solution, but the feasibility of active GCAS in these flight regimes may also be explored.

Investigate solutions to GCAS problems observed during engineering and piloted evaluations in stalled and near-stalled portions of the flight envelope. One possible answer may be for GCAS to add enough power to sufficiently elevate the aircraft's energy state prior to recovery initiation in order to avoid a dynamic or deep-stall condition.

Use simulation to correct the current inability of the active GCAS to effectively handle large asymmetric store loadings.

Develop a method to predict the average velocity of the recovery for use in the calculation of the dive angle compensation term. This term should reduce the amount of scatter in the recoveries due to the different acceleration profiles.

The roll time constant gain, K_2, should be varied with store loading in order to compensate for reduced roll rates resulting from FCL limiting with heavy store loadings.

Use simulation to investigate the effect of insidious INS failures and INS dumps on GCAS performance. Simulation is ideally suited to explore the repercussions of such failures and help point the way toward any necessary GCAS failure modes. Additionally, simulation may be used to safely evaluate the effects of degraded FCS modes on GCAS performance.

Use simulation to explore various switching and blending schemes between MSL and AGL sensors in order to compensate for the current failings in radar altimeter coverage.

Expedite installation of 360°-coverage radar altimeters onto production tactical aircraft.

Using the simulation, implement and evaluate the changes recommended by the pilots concerning the timing and amount of power applied by GCAS during recoveries. A long-term solution to the issues of ATC reliability and engagement ability could be the use of a digital engine controller.

Present the break "X" cue on all cockpit displays, so that the pilot may receive the necessary cuing in instances where he is watching a forward-looking infrared or strike camera image on a digital display indicator rather than the HUD. Use of the MFS motion platform and its virtual image mirror may allow more accurate evaluation of passive GCAS.

Perform a closed-loop analysis on the GCAS FCL's. This must be done prior to flight testing.

Implement the v1.1 GCAS in an F/A-18 technology demonstration airplane for flight testing. This may be done prior to completion of much of the previously recommended simulation work.

Finally, expedite installation of GCAS into production F/A-18 aircraft.

Acknowledgments

The authors would like to express their gratitude to members of the Simulation and Control Technology Department of the Strike Aircraft Test Directorate, and the Computer Technology and Simulation Department of the Systems Engineering Test Directorate for their contributions to this project. The authors would specifically like to thank Mark Hendrix and Michael Colony, who laid the initial groundwork for the F/A-18 GCAS algorithms presented in this paper; William McNamara and Cynthia Wathen, who provided engineering support for this effort; and the test pilots who participated in the evaluations: LtCOL Troy Pennington, USMC; LCDR James Galanie, USN; CAPT Christopher Hadfield, CAF; and LT David Prater, USN.

References

1. Skoog, M. A. et al: "AFTI/F-16 AMAS Ground Collision Avoidance System Evaluation"; AFFTC-TR-87-11, August 1987.

2. O'Bryon, J. F.: "Unlocking G-LOC"; Aerospace America; Vol. 29, No. 9, September 1991; pp 60-63.

3. Skoog, M. A. et al: "SFARS: Straight-Forward Autorecovery System"; unpublished, 1986.

4. Howard, J. D. et al: "AFTI/F-16 Gravity-Induced Loss-of-Consciousness and Spatial Disorientation Auto-Recovery System"; February 1986.

5. Howard, J. D. et al: "Testing Automated Ground Collision Avoidance Systems on the AFTI/F-16"; April 1986.

GCAS Implementation Schematic

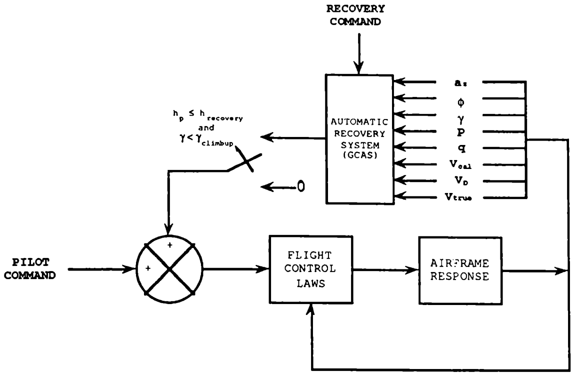


Figure 1

F/A-18 SFARS Implementation

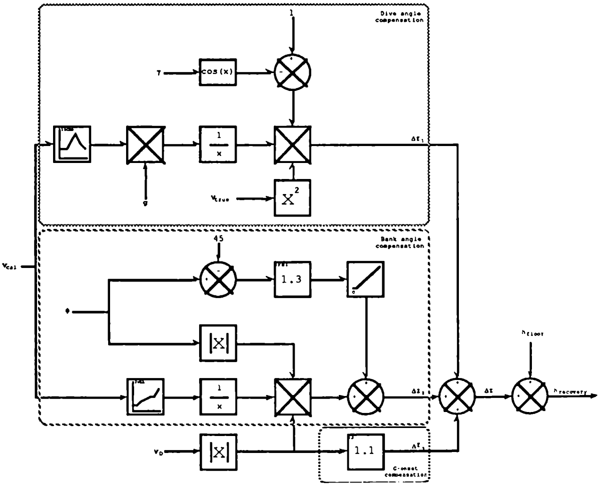


Figure 2

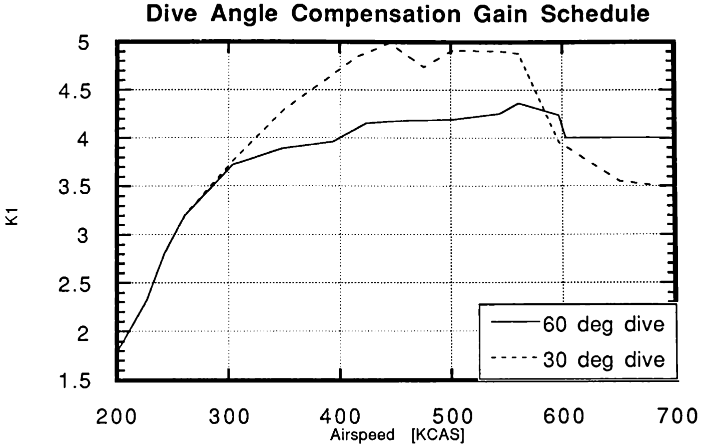


Figure 3

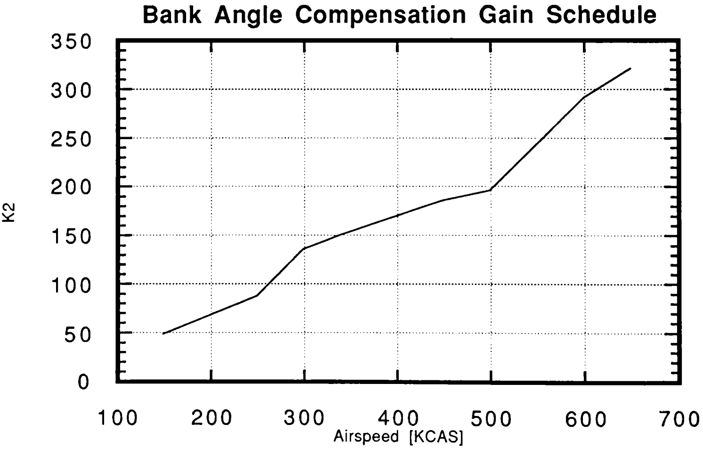


Figure 4

F/A-18 GCAS Control Law Additions

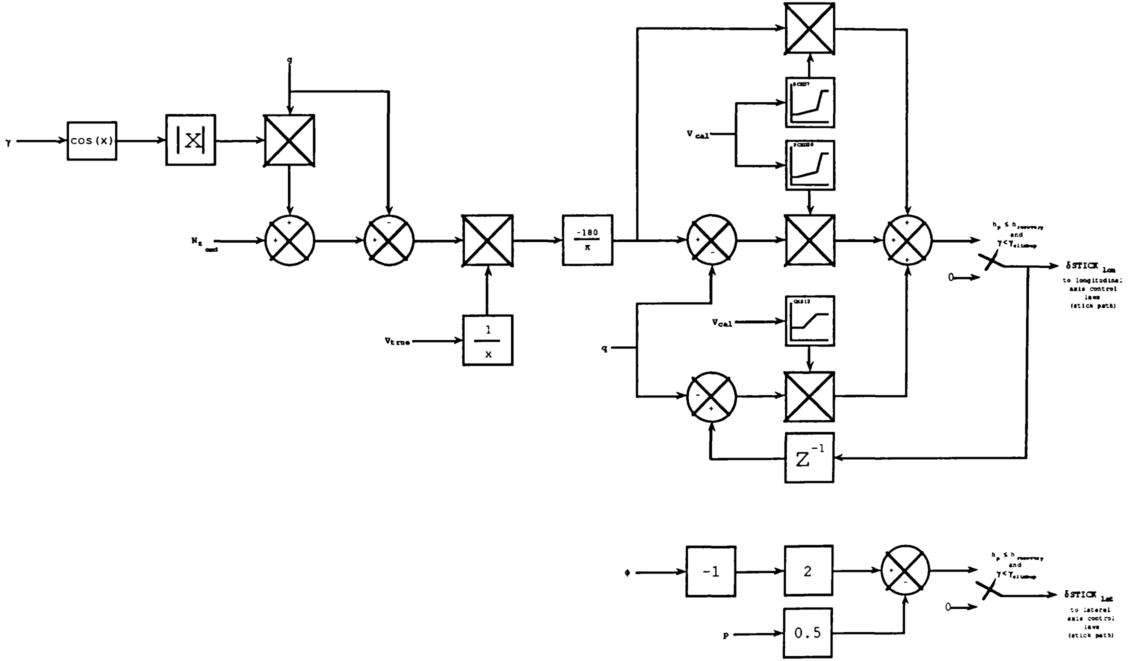
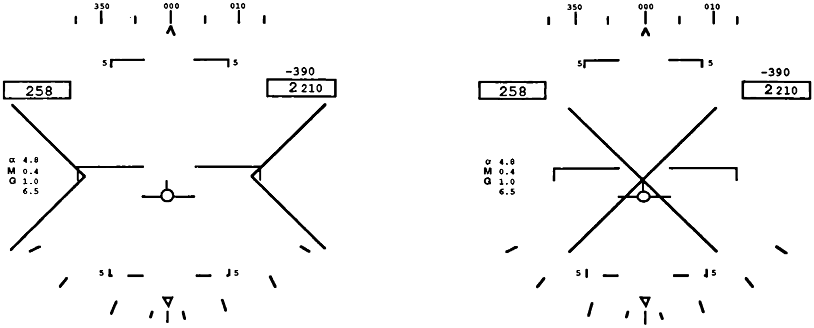


Figure 5

F/A-18 GCAS HUD Symbology

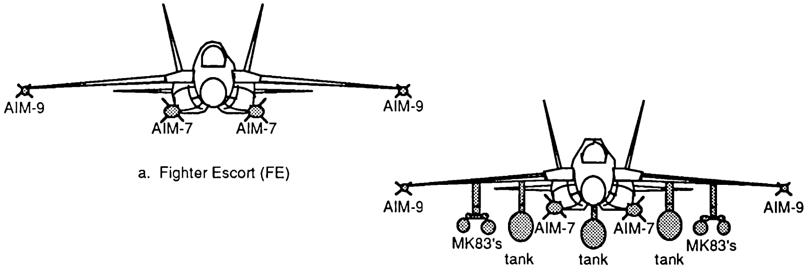


a. Chevron Presentation (~5 seconds prior to pull-up)

b. Break "X" Formation (at pull-up)

Figure 6

Tested Aircraft Configurations



a. Fighter Escort (FE)

b. Interdiction (INT)

Figure 7

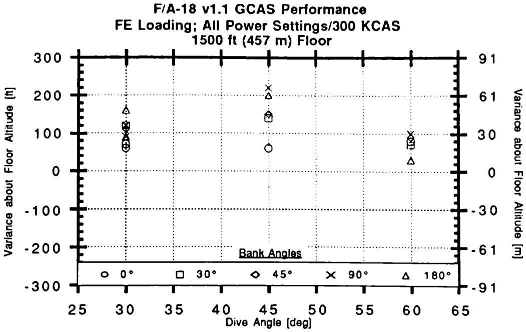


Figure 8

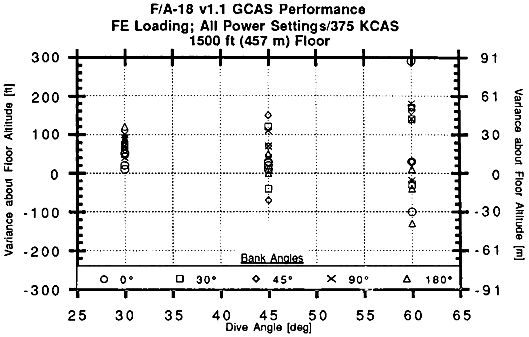


Figure 9

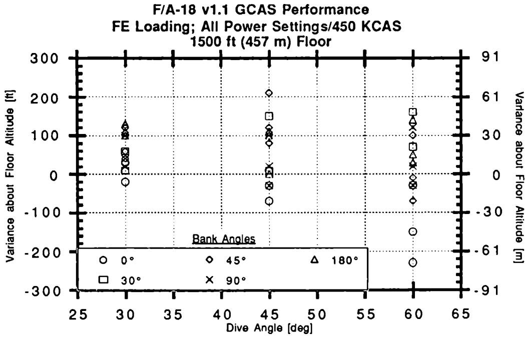


Figure 10

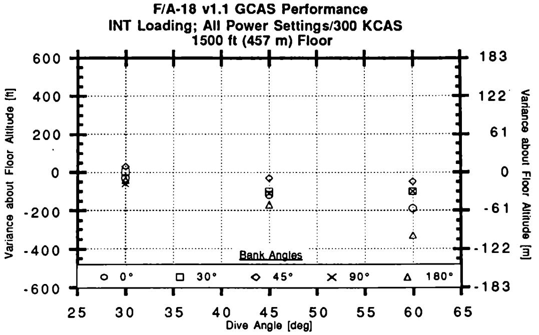


Figure 11

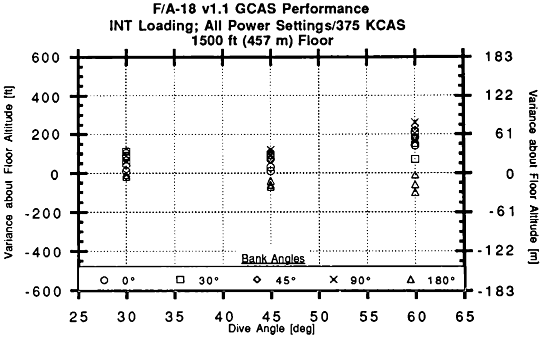


Figure 12

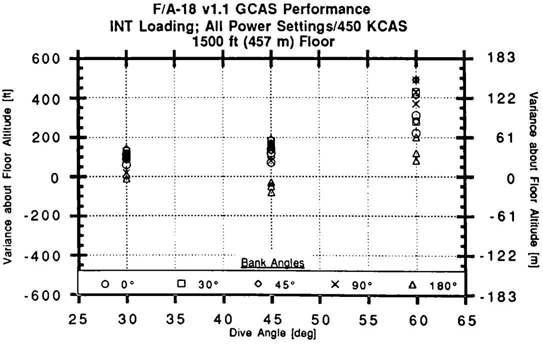


Figure 13

AIAA-92-4141-CP

USE OF SIMULATION TO PROVE MILITARY WORTH OF ADVANCED PLATFORM TECHNOLOGIES

F. McQuillan and R. Ure
CAE Electronics Ltd.
St. Laurent, Quebec, Canada

ABSTRACT

Spiralling costs in the development and evaluation of new military aircraft are forcing many agencies to look to simulation as a mechanism for advanced platform evaluation.

This paper describes a simulation facility being developed at CAE Electronics Ltd. to support a study of advanced VTOL concepts being conducted by the U.S. Army.

The facility allows researchers and engineers to evaluate the effectiveness of aircraft in performing selected missions. It provides a user-definable, high fidelity threat environment with terrain interaction. The experimenter can specify all attributes of the various players and the tactical scenario in a flexible and user-friendly manner. A data recording capability is available to log mission results for off-line analysis.

For the more promising airframes identified, a real-time piloted simulation is anticipated as the next phase of this project.

INTRODUCTION

The development and evaluation of new aircraft platforms for the military is an extremely expensive and time-consuming task. Given the recent advances in the sophistication of tactical simulators for training, the question arises as to whether these simulators could be extended or modified in some way to help streamline new platform evaluation. And if so, what form should such a simulation facility take?

The VTOL Effectiveness in Combat/Tactical Regimes (VECTR) project is an exploratory development project within the U.S. Army aimed at evaluating the military worth of advanced VTOL platforms. Current concepts under consideration are:

1. helicopter
2. compound helicopter
3. tilt rotor
4. tilt wing
5. tilt duct
6. fan-in-wing
7. jet lift.

The VECTR concept evaluation task consists of both analytical and piloted combat simulation. Analytical simulation implies that all battlefield participants, including the aircraft concept of interest, are controlled by a set of algorithms and not interactively by a human being. Such a simulation need not necessarily run in real time. Piloted simulation, on the other hand, occurs in real time with the pilot an active participant in the scenario.

This paper describes the Scenario Analysis for VTOL Vehicles using an Interactive Environment (SAVVIE). SAVVIE is a workstation-based analytical combat simulator. This work is being performed by CAE Electronics Ltd. as part of their contribution to the VECTR project.

VTOL AIR COMBAT

Before presenting the SAVVIE design, it is perhaps instructive to look at some of the areas that researchers might be interested in studying with such a simulation facility. This is not intended to be a primer on VTOL air combat, but rather just to indicate some of the issues surrounding the low level warfare debate.

Unlike fixed wing air combat which has a longer, well-documented history, doctrine for rotary wing air combat is written more or less in the absence of real-world experience (i.e., engagements between helicopters have rarely ever been reported). It is only in the last 20 years or so that systematic attempts have even been made to develop it. This uncertainty raises some basic questions for the aircraft designer regarding the specific maneuvering and combat environment to design for.

Rotary wing aircraft first appeared on the battlefield about 30 years ago in the form of the single rotor helicopter. Its primary attributes were efficient hover, good low speed maneuverability, low downwash, low empty weight and symmetrical yaw control. And though many advances in helicopter technology have been made since that time, a significant list of deficiencies still remains: low maximum speed, low range, limited high speed maneuverability, high vibration and coupling of motion degrees of freedom.

Consider the coupling problem, for example, which is an important factor affecting pilot workload and hence aircraft agility. Helicopters have only four cockpit controls to control six degrees of freedom. Airspeed and pitch are coupled, as are lateral speed and roll (at low speed) or yaw rate and roll (at high speed). Advanced VTOL vehicles tend to be more symmetrical in their response to control inputs and thus attitude control can be decoupled from airspeed to some degree. This is just one of the reasons why investigation of new VTOL platforms is justified.

Some of the recent VTOL concepts such as the jet lift and fan-in-wing have a much higher maximum speed (dash) capability than the helicopter. There are those who assert that exploiting this speed can lead to a significant combat advantage and thus high energy maneuvering should form the basis of rotary wing air combat. Others, however, maintain that the low detectability of the nap of the earth (NOE) rotary wing represents its most effective characteristic. Low speed maneuverability, agility and low signature should not be sacrificed in the pursuit of pure speed. Fast turn and shoot will be the order of the day, not complex maneuvering for firing position as is the case in fixed wing air combat.[1]

Figure 1 from Reference 2 suggests the trend that may evolve when evaluating a VTOL candidate platform on an offensive mission. The probability of an enemy kill depends on the individual probabilities of the weapon systems encountered. What speed should the aircraft maintain to maximize survivability? For which scenarios would this speed be beyond the capability of current helicopter designs?

Next consider maneuverability as a function of airspeed. Figure 2 from Reference 1 compares the turn rates of a typical helicopter such as the AH-64 with a tilt rotor such as the XV-15 or V-22. Turn rate can be thought of as a measure of how fast the aircraft can turn to direct its weapons towards an opponent, such as the "Copter killer airplane" indicated on the same figure. Below 110 knots, the modern helicopter has an advantage in this regard, but in the chaotic world of air combat where turns may be performed at a range of speeds above and below 110 knots, where does the advantage really lie?

SAVVIE offers a complete simulation environment with realistic threat modelling and terrain interaction to permit a thorough investigation of these and other issues.

SAVVIE FUNCTIONAL REQUIREMENTS

The goal of SAVVIE is to permit an experimenter to easily create scenarios in which a VTOL vehicle can perform a mission in the presence of friendly and threatening forces. The scenarios must faithfully replicate the actual combat environment from the point of view of terrain, weather, dynamic characteristics of the participants and armament.

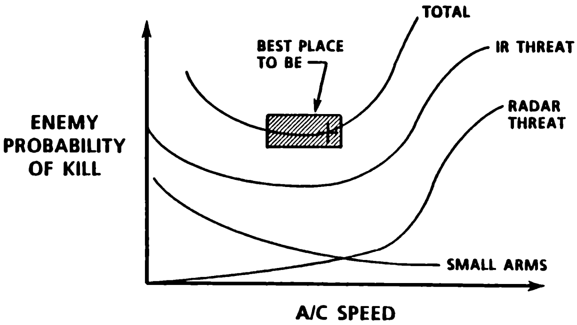


Figure 1. VTOL Vehicle on Offensive Mission (From Reference 2)

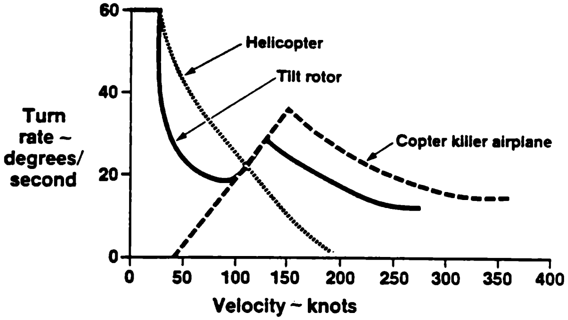


Figure 2. Turn Rates for Various Configurations (from Reference 1)

Reference 1 makes several specific suggestions in this regard:

> "The detail of the terrain base must closely match the size of the helicopter, so that careful intervisibility calculations can support detectability and survivability trades. The rules which apply to battlefield conduct of the NOE helicopter must be understood and applied in the battle simulation. Accurate representations of flight profiles, speed and altitude relationships and threat sensor performance in clutter must be input so that realistic answers are obtained."

With regard to flight modelling, a simple stability derivative model or a high-fidelity model could equally be used depending on the nature of testing to be performed. Therefore, the design of the system should be such that flight dynamics models for VTOL vehicles may be added at any time and easily integrated by the experimenter. This demands a general and well-defined interface between SAVVIE and the library of external aircraft models.

The hardware configuration for SAVVIE consists of a single computer system with a graphic console and terminals, laser printer and joystick (Figure 3).

The main software components of SAVVIE are:

1. a data base management system (DBMS) to define the experiment
2. a simulation package to run the experiment
3. a set of graphic tools to observe and control the experiment
4. a data recording system to gather and store experimental results for later analysis.

The data base structure and run-time software are existing modules that comprise the Interactive Tactical Environment Management System (ITEMS). ITEMS is a real-time, knowledge-based software product developed at CAE to construct and execute tactical scenarios. It is currently being used in a number of simulation applications and is described in more detail elsewhere.[3,4]

The following sections will consider each of the four main modules listed above. Many of the features of ITEMS will not be covered in this paper so as to focus primarily on the VTOL analysis capability of SAVVIE.

DATA BASE MANAGEMENT SYSTEM

The DBMS is a design and preparation utility enabling the experimenter to completely define a scenario, having little or no computer science background. DBMS is based on the X Window System and features a Graphical User Interface (GUI) that is easy to use. On-line help and validation of user inputs before execution are provided.

A scenario may contain up to 100 players, a player being a vehicle with its equipment and dynamic characteristics, along with a mission plan and doctrine. The main player in the SAVVIE scenario is the VECTR player representing the VTOL vehicle under investigation. A maximum of two VECTR players per scenario may be defined.

DBMS is a hierarchical data base with unidirectional links between elements. Higher level data bases refer to lower level ones which contain more detailed information about a particular element.

Refer to References 3 and 4 for a description of DBMS

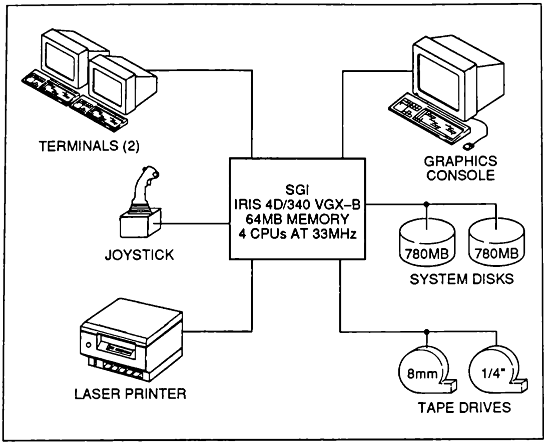


Figure 3. SAVVIE Hardware Configuration

libraries. Here we consider only the doctrine contained in the rules data base.

Rules Data Base

ITEMS uses rules to encode players with knowledge that permits them to interact intelligently with other players, and also permits non-player systems to perform their functions without operator intervention. ITEMS contains a rules editor for entering knowledge and a run-time inference engine for processing it.

The basic knowledge primitive in ITEMS is a production rule of the form:

IF (condition) THEN (consequence)

A military tactician or experimenter can create a series of rules which are subsequently processed by the ITEMS run-time software. This doctrine may be general or may be associated with a particular scenario.

There are two classes of doctrine: action-oriented and selection-oriented. Action oriented doctrine contains rules that control the actions of a mission or VECTR player. The inference engine uses a simple mechanism to schedule them based on placement of the rules in the database. This class of doctrine is well-suited for encoding standard operating procedures such as weapon selection and basic maneuvering. An example of an action-oriented rule is:

IF (player radar tracked)
AND (RF jammer available)
THEN (turn on RF jammer)

Selection-oriented doctrine provides a mechanism for selecting the best set of objects from a larger set of similar ones. Instead of defining one condition in which an outcome will be definitely true, selection-oriented rules define a number of conditions in which the outcome will be true to different degrees. In SAVVIE, selection-oriented doctrine is useful to determine which threat to attack out of the set of all detected threats. An expert system primitive called a certainty factor is used to implement this.

A certainty factor is a real number between 0 and 1. It is a quantitative indication of how certain the tactician is about a particular selection. A certainty factor of 0 indicates 100% certainty that something should not be selected, while a certainty factor of 1 expresses complete certainty that it should be selected. The value 0.5 indicates neutrality. An example of a selection-oriented rule is:

IF (threat is a helicopter)
THEN (set certainty 0.75)

```
IF   (closing speed greater than zero)
THEN (set certainty 0.8)

IF   (closing speed less than zero)
THEN (set certainty 0.2)

IF   (threat is closer than other threats)
THEN (set certainty 0.9)
```

Every player has action-oriented doctrine for controlling its mission and selection-oriented doctrine for deducing the most suitable prime opponent. The VECTR players in particular also have action-oriented doctrine for air combat and data recording.

RUN-TIME SIMULATION SOFTWARE

The run-time facility permits the experimenter to control and monitor scenarios that have been created through the DBMS editor. The main modules are shown in Figure 4 and are briefly described in the following sections.

Scenario Manager

The Scenario Manager is responsible for activating and deactivating mission players. The concept of active and inactive players exists so that a scenario may contain more players than can be actually processed with the available CPU power. Inactive players receive minimum attention; they move along their mission routes but cannot detect or cannot be detected by other players. Active players receive the bulk of the processing bandwidth allowing them to interact with each other in a realistic manner.

Current limits are two VECTR players, 44 active and 100 total mission players, though it is possible to adjust these limits subject to hardware limitations or degradation in execution time. The state of a particular player is continuously evaluated to determine if it should be acive or inactive according to its range from the VECTR players.

The Scenario Manager also deactivates players if they have been hit by a weapon or collide with another player, the ground or a terrain feature. The experimenter has some override capability with regard to player status; he may force activation or deactivation, recover a killed player or put players in kill-inhibit or collision-inhibit mode.

Player Controllers

Mission Controller. The Mission Controller is used to control both VECTR and mission players and comprises two phases. In the first phase, all mission players detected by the sensors are scanned and the prime opponent selection doctrine is used to select one of these players as the prime opponent. As discussed above, the rules that comprise this doctrine express how to evaluate an opponent in terms of threat level and target opportunity. A mission player may only have one prime opponent at any given time.

In the second phase, a plan of action is formulated by the inference engine. The inference engine will consider all information gathered about the prime opponent through a player's sensors, as well as the player's own flight parameters and munitions.

The Mission Controller has three ways to control maneuvering. The first way is to use waypoints. A player will visit each of its waypoints in the manner set out by the experimenter during definition of the mission route.

Formations may also be used, whereby a series of players are slaved to a master. When a player engages an opponent, the Mission Controller removes it from the formation and controls its maneuvering. After the engagement, the slaved player will return to its position in the formation.

The third maneuvering system is the action doctrine itself. Through the doctrine, the experimenter defines the conditions under which a player stops following its mission route and performs another action. This may entail approaching an opponent or performing an evasive maneuver.

The plan of action formulated by the inference engine is based on knowledge embedded in the player rules and information gathered on the prime opponent through player sensors. Like real-world sensors, the simulated sensors have limitations that may cause them to lose track of an opponent during an engagement.

Air Combat Controller. The Air Combat Controller is used to control VECTR players for high fidelity air-to-air and air-to-ground interactive combat. The experimenter is free to design different sets of air combat rules through DBMS for each of the VTOL platforms.

One part of the Air Combat Controller is a virtual environment that computes various air combat parameters such as range, angle off tail, closure rate and the like between a player and its prime opponent. The second part is the inference engine which decides which maneuver the VECTR player shall perform based on its current state and the state of the tactical environment. For example, the inference engine may decide that the VECTR player should track the prime opponent and use its main gun.

Interactive Navigation. Interactive Navigation receives inputs from the Mission Controller and writes outputs to Vehicle Dynamics. It directs air and ground vehicles to move and interact with the terrain according to navigation mode. There are six navigation modes:

1. Direct Mode. Flies an airborne player from point A to point B in a straight line at a certain altitude above sea or ground level. Generally used for high altitude flight.

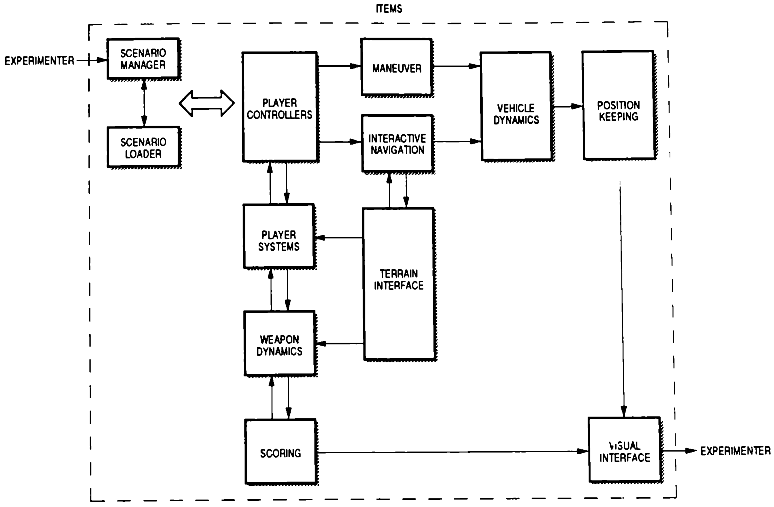


Figure 4. ITEMS Functional Diagram

2. Low Level Flight. Flies an airborne player to clear the highest point along the flight path using a look-ahead time.

3. Contour Flight. Similar to low level fl ght, but with a shorter look ahead time so the terrain is followed more closely. Generally for low speed, low altitude flight.

4. NOE Flight. Flies an airborne player very close to the ground and take advantage of natural or man-made obstacles to stay masked from potential threats. Characterized by low speeds and large lateral movements.

5. Obstacle Avoidance. Selects a path around obstacles to reach a waypoint using a simple, non-recursive algorithm. Designed for use by ground vehicles in lightly cluttered environments.

6. Road Following. Drives a ground vehicle along a road segment. Obstacle avoidance is also active during road following.

Player Systems. Player Systems are defined as systems contained on a player's platform. Current simulated systems include:

1. active sensors
2. passive sensors
3. countermeasure systems
4. weapon systems
5. laser systems
6. other systems such as fuel and turrets

Consider the laser system for example. Lasers can be defined to function as either range finders or as designators. Range finders accurately measure the distance between a player and a specific target, whereas designators identify a spot on or near a target to direct a laser-guided missile.

The DBMS utility can be used to define the characteristics associated with each laser type. For instance, a designator system can have its beam encoded with different laser codes and the designating envelope specified in azimuth and elevation. During laser system processing, the command designate prime opponent target will attempt to keep a designation mark on the players prime opponent. Successful execution will depend on the particular laser having designating capability, the target remaining within line of sight and, say, the target not being hidden behind a cloud of smoke.

Enhanced Dynamics

Overview. The Enhanced Dynamics module controls the VECTR model assigned to a VECTR player. The principle elements are the Maneuver Module, Autopilot and Control Mixing, Interface Module and Aircraft Dynamics (Figure 5). Definition of the airframe is done in DBMS.

Maneuver Module. Based on the current tactical situation and the doctrine that has been defined, the Player Controller sequences a series of maneuvers to be processed by the Maneuver Module. These maneuvers encompass offensive and defensive air-to-air and air-to-ground engagements. The Maneuver Module then computes the appropriate velocity, attitude, altitude and other demands to be sent to the autopilot.

Each maneuver comprises an initialization and execution component to ensure smooth transitions. Some maneuvers are meaningless as stand alone actions and it is only upon concatenation of two or more of them that a realistic flight trajectory is constructed. For example, a climb maneuver trades speed for altitude. At some point the Player Controller must stop the climb and demand, say, a roll to remove relative bearing. A prolonged climb could result in loss of too much speed and associated lift.

Various basic maneuvers such as pursuit, climb, dive, low yoyo and hover are supported, as well as other more complex ones such as dive-break and pitch-back attack.

Autopilot and Control Mixing Module. Here the control loop is closed around the aircraft state as commanded by the maneuver module. It outputs the pilot control deflections to the flight dynamics module. For the helicopter, stick controls are lateral cyclic, longitudinal cyclic, collective and pedal displacements.

Some control mixing is required for different types of VTOL vehicles. For the tilt wing aircraft, for instance, airspeed would be controlled by tilting the rotor during low speed flight and using the engine power during high speed flight.

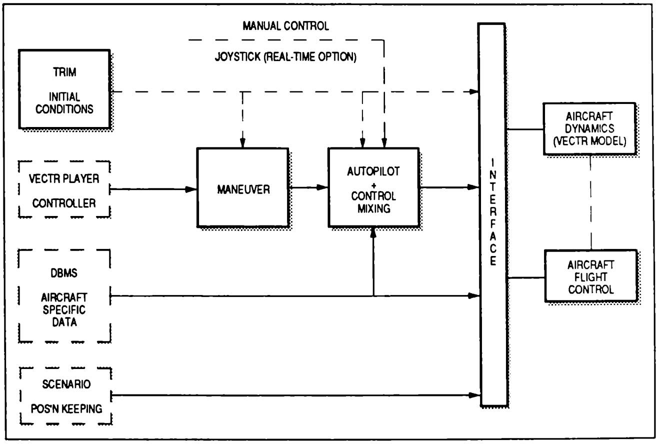


Figure 5. Aircraft Operational Flow Diagram

Interface Module. The Ir.terface Module provides a general purpose mapping between the SAVVIE environment and the VECTR model. In the SAVVIE to VECTR model direction, the data transferred is:

1. database information defining the airframe
2. demanded aircraft state
3. aircraft position
4. terrain and ambient conditions

The only data passed from the VECTR model back to SAVVIE is the computed aircraft state vector.

Aircraft Dynamics. This module consists of the library of VECTR models that may be plugged in to investigate platform performance in some scenario of interest. These models may be of any complexity at all, as long as they output the aircraft state back across the interface.

An in-house six degree of freedom single rotor helicopter model is being used as a benchmark for development and proof of performance. Other VTOL platforms are to be supplied and integrated by the user as they are developed.

Simple Dynamics

Helicopter, fixed-wing and ground vehicle mission players are represented by simple dynamic models. These are computationally efficient yet still give realistic responses to speed, altitude and heading demands.

A point mass approximation is used with no external forces considered to be acting on the airframe or ground vehicle. Response to input commands is first order.

Weapon Dynamics and Scoring

Weapon Dynamics simulates missiles, rockets and bullets fired from a VECTR or mission player. The two primary classes are ballistic projectiles and guided projectiles. Ballistic projectiles (rockets and bullets) employ a three degree of freedom model with gravity effect. A stream algorithm is used for flight by packet, within some defined dispersion cone, since there may be many in flight at a given time. Guided projectiles (missiles) are controlled in one of two modes. Proportional navigation (PN) is used for missiles carrying their own on-board tracking system or homing head. Acceleration demand to the missile is based on rate of change of the missile/target line of sight. Command to line of sight control (CLOS) is used for missiles that are tracked and guided from an on-ground tracking system. Here the acceleration demand is proportional to the differential angle between the radar/target line of sight and the radar/missile line of sight as perceived by the ground tracking system. Infra red, active radar and laser homing heads may be used, with consideration given to seeker field of view, gimbal limits and gimbal rate limits. Input to the homing head is the detectable energy from the target and any counter measures that may be employed: chaff, flares, smoke or jammers.

The scoring algorithm detects a hit or miss condition when a weapon is released. Degradation of performance is determined by the severity of damage to the target.

Terrain Interface

The Terrain Interface allows SAVVIE to interact with the terrain. It contains line of sight computations and extraction of data used for player navigation. The gaming area residing in memory consists of three windows of different sizes and resolutions (Figure 6):

1. scenario low resolution database (100 km X 100 km)
2. scenario high resolution database (32 km X 32 km)
3. player terrain database (4 km X 4 km)

The scenario low and high resolution data bases contain coarse and fine terrain bit-map representations respectively. Both also describe feature type and feature height.

The player terrain database consists of polygons which detail the ground height and slope in the immediate vicinity of a player. This ensures realistic motion of ground track vehicles and flight of airborne players at low altitudes.

VISUAL AND TERRAIN DATABASES

All tactical scenarios developed for SAVVIE utilize the Fulda Gap terrain database, which covers a 106 km by 84 km area in Northern Germany containing a variety of geographic features. At the centre of the database is a higher resolution area, approximately 30 km by 25 km, which contains a number of small villages, power lines, vegetation and detailed terrain information. The original visual database files are used to extract data which is then reformatted for an Iris-based display.

The two main SAVVIE visualization tools are the Forward View Display (FVD) and the Tactical Scenario Display (TSD). The FVD offers a three-dimensional, out-of-cockpit view from any player in the scenario. The operator may look around, set angular and linear offsets from a player or adjust the field and depth of view. The view itself consists of ground polygons covered with natural and cultural features and is updated at a near real-time rate.

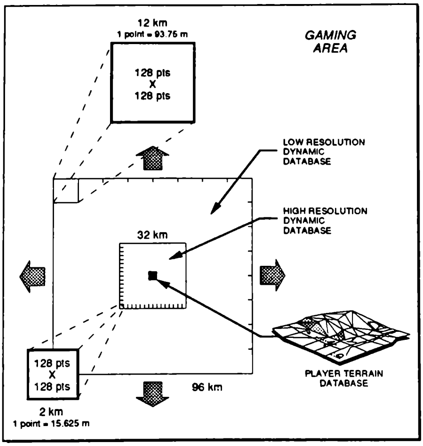


Figure 6. ITEMS Terrain Data Bases

The TSD provides a god's-eye view of the evolving tactical scenario. It may be centered on any player or at a random position in the database. Under operator directions, the display may include contour lines, roads, rivers, lakes, forest areas, urban areas and a coordinate grid. Tactical symbology is userdefinable through DBMS.

DATA RECORDING

SAVVIE includes a data recording capability which allows the experimenter to select the type and level of detail of data to be acquired during scenario execution. Data collection is event-based, where an event is defined by occurrence criteria. A series of associated environment variables define what to collect at the instant of the event.

There are three recording modes:

1. One shot. One-time recording of data at the instant a condition is met.

2. Continuous. Record data at userdefined frequencies.

3. Triggered. Record data at userdefined frequencies while a set of conditions is true.

Composite variables derived from mathematical or boolean operations on basic recording parameters are also supported.

Data recording rulesets stored in DBMS are processed by the inference engine which determines the variables to store. For example:

IF (condition block VECTR UNMASKED true)
AND (vectr player speed less than 200 m/s)
THEN (record the list TELEMETRY in one shot mode)

Condition block VECTR UNMASKED might look like:

(line of sight is true)
AND (range to enemy less than 5 km)
AND (enemy facing vectr player)
AND (...)

The list TELEMETRY would detail the variables to be logged.

An extraction module converts saved data from binary form to tabular form suitable for standard data analysis tools. All output is written directly to 8 mm cassette tape.

SUMMARY

SAVVIE provides the capability for analytical evaluation of the military worth of advanced VTOL platforms. The experimenter can specify all attributes of the various players and the tactical environment in a flexible and user-friendly manner. Doctrine specifying rules of engagement is entered through the DBMS rules editor and an inference engine processes it at run-time to give realistic player interaction. The terrain detail is sufficient so that NOE warfare may be accurately simulated. Flight dynamics models of varying complexity for the VTOL platform may be added at any time without the need for the manufacturer to modify the basic simulation software. A high level of security is therefore assured as the tactician and engineer have complete control over the experiment setup and execution. During testing, graphical displays of the evolving tactical scenario are available. A rule-based data collection facility is also provided to log experiment variables for later off-line analysis.

REFERENCES

1. Lappos, N., "Designing for Helicopter Air Combat," Conference on Air-to-Air Combat for the Fighter Helicopter, January 1990.

2. Magee, J.P., "Designing for the Low-Level Air War," Conference on Air-to-Air Combat for the Fighter Helicopter, January 1990.

3. Morris, A., "The Creation of Complex Tactical Training Environments, An Unclassified Approach," 10th Interservice/Industry Training Systems Conference, pp. 151-160, November 1988.

4. Ure, R. and Siksik, D., "User Definable Doctrine for Interactive Air Targets," AIAA Flight Simulation Technologies Conference, pp.359-363, September 1990.

ABBREVIATIONS

CLOS	Control to Line-of-Sight
DBMS	Database Management System
FVD	Forward View Display
GUI	Graphical User Interface
ITEMS	Interactive Tactical Environment Management System
NOE	Nap-of-the-Earth
SAVVIE	Scenario Analysis for VTOL Vehicles Using an Interactive Environment
TSD	Tactical Situation Display
VECTR	VTOL Effectiveness in Combat/Tactical Regimes
VTOL	Vertical Take-Off and Landing

AIAA-92-4143-CP

CONTROL BUS MODELING IN THE SHUTTLE MISSION SIMULATOR

William B. Miller, III*
Paramax Systems Corporation
Houston, Texas 77058

Timothy R. North
Rockwell Space Operations Company
Houston, Texas 77058

Abstract

Each space shuttle is equipped with nine electrical control buses in the flight deck and middeck areas of the crew compartment. NASA/Johnson Space Center Mission Operations flight controllers developed malfunction procedures to be used by astronauts and flight controllers in the event of a control bus failure during a shuttle mission. The results of engineering tests performed to validate these procedures showed that all control bus failure modes were not adequately addressed. This meant that both the malfunction procedures and the existing Shuttle Mission Simulator models needed to be revised. Extensive modifications were needed so that the models could be used to train astronauts and flight controllers on the new procedures.

The new software written to support these modifications consisted of nine new interactive displays written in Aydin Page Source Language and two blocks of Univac FORTRAN 11 code added to the existing shuttle electrical power system simulation routines. All simulation models using control bus power had to be modified to reference the status of the correct control bus and panel.

This paper describes how the test results drove the simulation requirements and how the resulting math model was designed and implemented.

Introduction

The Shuttle Mission Simulator (SMS) is the prime simulator for crew training in the shuttle program. It is the only high-fidelity simulator capable of training flight crews for all phases of a shuttle mission. The SMS can be integrated to the Mission Control Center (MCC) via the Network Simulation System (NSS) to provide simultaneous training capability for flight controllers and crews in a simulated flight environment. The SMS complex consists of three simulation stations with one Univac 1182 host computer and two Concurrent Computer Systems 3280 intelligent controllers per station, plus digital image generation equipment[1].

Each space shuttle orbiter vehicle is equipped with nine electrical control buses in the flight deck and middeck areas of the crew compartment. These buses serve to provide direct current (DC) power to display and control panel switches and associated logic, rather than to provide operational power to any subsystem. Each control bus is made up of approximately 500 feet of 22 and 24 gauge wire and is strung from switch to switch behind the control panels. Many critical orbiter subsystems utilize logic signals driven by control bus power, including the Main Propulsion System, Orbital Maneuvering System, Reaction Control System, Auxiliary Power Units and Hydraulic System, Electrical Power System, Environmental Control and Life Support Systems, Landing and Deceleration Systems, Communications Systems, and the Data Processing/Navigation Systems.

Each of the nine control buses is redundantly powered from all three of the orbiter's main DC buses. Two of the main DC buses are connected to the ends of

* Software Engineering Supervisor
Senior Member AIAA

the control bus through remote power controllers (RPCs). An RPC is a solid-state switching device used to protect against high electrical currents. The RPC will limit the current passing through it to 150 percent of its rated value. If the high current persists for approximately 2.5 seconds, the RPC will trip off and remove the output current. A tripped RPC may be reset by a crew member by cycling the associated control switch. The RPCs used in the main-to-control bus connection are rated for 5 amperes. The remaining main DC bus is connected near the center of the control bus through a 5-ampere fuse in series with a 10-ampere circuit breaker. Each control bus is equipped with a voltage sensor located near an RPC at one end of the bus. Figure 1 shows a schematic of a representative control bus and the display and control panels through which it passes[2].

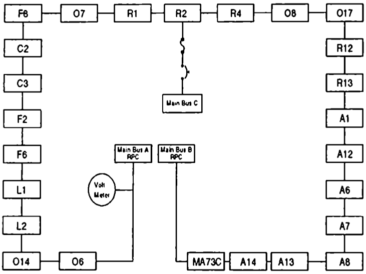


Figure 1. - Typical Control Bus Schematic

Original Malfunction Procedures and Model

In the event of a short circuit on a control bus, it was believed that the RPCs on each end would trip and that the fuse in the center would blow. Procedures were written to take advantage of the RPC's characteristic 2.5-second time to trip off. If a control bus tripped off, it was thought that a crew member could hold an affected switch to the desired position and then reset the corresponding RPCs. This became known as the "three finger trick".

This concept of control bus operation was carried over into the SMS. A simplistic model was implemented which assumed a "lumped" control bus. Nine logical flags, one per control bus, were set by the electrical power system simulation model and referenced by all subsystems utilizing control bus power. Malfunction capability in this control bus model included the capability to fail the RPCs and fuses, and the ability to simulate a short by applying an extra load to the bus. A total load on the bus was calculated which included both normal loads and loads due to these short malfunctions entered by the instructors. If this total load exceeded the combined limits of the RPCs and fuse, the RPCs tripped and the fuse blew. The appropriate control bus availability flag would then be set FALSE. All subsystem models utilizing that particular control bus would then respond accordingly.

Flight controllers, wishing to get confirmation that the "three finger trick" would work in an actual mission situation, requested that engineering tests be performed to validate this procedure.

The Tests and Results

A breadboard of each control bus was constructed by the Engineering Directorate Avionics Systems Division at Johnson Space Center. Control bus re-power tests were performed with shorts of varying magnitudes placed at various locations along each bus[3].

The results indicated that resetting the RPCs had varying degrees of success dependent upon the magnitude and location of the short. The resistance and length of the wire constituting the buses causes voltage gradients along the length of a shorted bus. The voltage at a particular bus location is proportional to the distance from the short. In some tests, one of the RPCs tripped. In other tests, one RPC tripped and the fuse blew. Finally, some tests did not trip either RPC or blow the fuse. No test case caused both RPCs to trip.

Figure 2 shows one set of data from the tests. This figure presents the bus voltage at various panel locations for five locations of a 0.1 ohm short. A line is shown at 9.35 volts, which is the approximate minimum voltage for operation of the logic devices powered by the control buses.The results in this figure are representative of all the tests performed. In general, panels in the immediate vicinity of a short would be heavily affected and perhaps nonfunctional. Panels further removed would be lightly affected, if at all. These results showed that some of the assumptions underlying control bus operation and the "three finger trick" were invalid. The results showed that for certain situations, even resetting the current-limited RPC would not be sufficient to raise the control voltage enough to control a switch load.

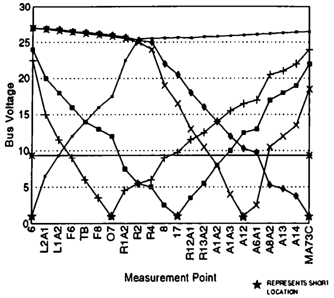


Figure 2. - Representative Control Bus Test Results

Flight Rules and malfunction procedures would have to be changed in light of this new data. In addition, these results rendered the existing SMS control bus model obsolete. Extensive modifications were needed so that the models could be used to train astronauts and flight controllers on the new procedures.

New Malfunction Procedures and Model Requirements

New Flight Rules and malfunction procedures were written with the understanding that for any single failure, only a portion of the control bus would be lost. The procedures only allow reset of a single RPC in an attempt to regain a critical function. The flight crews no longer routinely attempt resetting RPCs[4].In addition, flight controllers and flight crews began thinking in terms of partial bus losses and how to identify which portion of a control bus was failed. Flight controllers planned to identify failed portions of the bus by monitoring switch position telemetry information. The existing control bus model could not support partial bus loss training scenarios.

The simplistic assumption in the existing control bus model that the bus could be treated as a "lumped" or point-like entity was the single largest constraint that prevented exercise of these new malfunction procedures. A segmented or distributed bus model needed to be implemented to allow for these partial bus failure scenarios. However, this immediately raised tradeoff issues about the level of fidelity needed versus the SMS computer complex resources, since the SMS computers are highly constrained in both available memory and execution time. Several solutions were considered.

The highest level of fidelity would be achieved by modeling the individual resistances of each control bus segment between each panel and computing current and voltage at each switch. This implied high impacts to memory allocation and execution time as well as implementation manhours. Alternately, the model could be left unchanged. An elaborate "work-around" scheme was devised by instructors that involved the manual malfunctioning of dozens of telemetry parameters affected by each shorted bus in order to give flight controllers the correct indications. This was extremely complicated and labor-intensive for the instructors and introduced a large potential for human error due to the large number of manual data entries involved. Any such error would give the flight controllers a flawed failure signature and could result in an incorrect problem diagnosis, invalidating the desired training goal.

The option that appeared to have the best tradeoff between implementation cost, computer resource impact, and user functionality was to replace the nine control bus availability flags with a matrix of flags whose indices represented the individual control buses and panels. The instructor would be able to malfunction any panel on any bus individually. The electrical power system model would set statuses in the matrix based on these instructor malfunction inputs and the status of the main buses feeding power to the control buses. All other models which required control bus power as an input would be changed to reference the correct entry in the new matrix instead of the old control bus availability flags. The effect of shorts on the control bus voltage sensor reading would not be calculated automatically, but could be manually controlled by the instructor.

Instructor Interface Considerations

Although this option appeared to be the only feasible one in terms of computer resources, serious operational considerations still needed to be addressed. The existence of up to 27 panels on any of the nine control buses implied 243 individual new malfunctions. Care needed to be taken to ensure that a workable scheme was devised for both easily inserting as many of these malfunctions as were desired and monitoring the results. Existing monitoring capability for control bus status was limited to displays showing the status of each of the nine availability flags. Using the existing SMS malfunction insertion system did not seem the best option. To insert a malfunction into the simulation the instructor must first individually select each malfunction from a set of menu pages. This causes the malfunction to appear on an insertion menu which is limited to 30 entries per instructor. At this point the malfunction must be selected again with the lightpen to insert it into the simulation. Utilizing this system would make multiple panel and bus failure scenarios operationally clumsy at best.

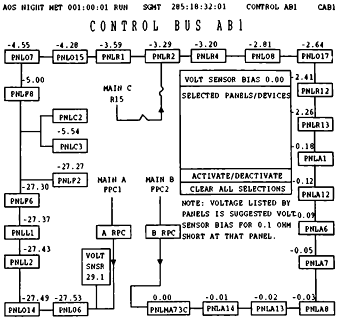
It was decided to address both the malfunction capability and statusing requirements by creating a set of displays providing an interactive schematic representation of each control bus and panel[5]. These displays would provide the following information:

1) The power availability status of each panel
2) The power availability status of the three main buses supplying the control bus
3) The operational status of the two RPCs
4) The operational status of the fuse
5) The position of the circuit breaker
6) The reading on the voltage sensor
7) All panels the instructor has selected to malfunction
8) Malfunction activation status
9) A suggested voltage sensor bias for each panel failure

The displays would provide interactive capability as follows:

1) The ability to select any combination of panels to be malfunctioned
2) The ability to malfunction the RPCs and fuse
3) The ability to bias the voltage sensor reading

The new displays were written in Aydin Page Source Language (a site-unique language) for execution on the SMS Base Intelligent Controller, a Concurrent Computer Systems MPS 3280. This computer handles many of the interfacing chores between the SMS host computers, crew station hardware, and other systems. Figure 3 gives an example of one of the completed displays. All panels and devices are shown in green if power is available and in red if power is not available. The reading on the voltage sensor is displayed in the box containing the heading "VOLT SNSR". The instructor can select all devices that are to be malfunctioned, including panels, RPCs, and fuses, by touching them with the lightpen. As each device is selected, it is listed in the box containing the heading "SELECTED PANELS/DEVICES". The instructor can enter a bias to the voltage sensor in the box labelled VOLT SENSOR BIAS. When all selections are complete, the instructor can select the field labeled "ACTIVATE/DEACTIVATE" with the lightpen. At this point all the requested malfunctions will be inserted into the simulation, the "ACTIVATE/DEACTIVATE" field will be displayed in reverse field, the voltage sensor bias will be applied to the voltage sensor reading, and all newly failed panels and devices will turn red. As long as the "ACTIVATE/DEACTIVATE" field is displayed in reverse field, other devices may be selected with the lightpen and will be immediately failed or restored to function, depending on their status prior to being selected. The only exceptions to this are the RPCs, which may be failed in this manner, but can only be restored to functionality by first removing the malfunction and then toggling the appropriate RPC reset switch. If the "ACTIVATE/DEACTIVATE" field is again selected, all malfunctions will be removed from the simulation and the field will be displayed in normal text.



AOS NIGHT MET 001:00:01 RUN SGMT 285:18:32:01 CONTROL AB1 CAB1

CONTROL BUS AB1

NOTE: VOLTAGE LISTED BY PANELS IS SUGGESTED VOLT SENSOR BIAS FOR 0.1 OHM SHORT AT THAT PANEL.

Figure 3. - Typical Control Bus Interactive Display

Electrical Power System Simulation Model Updates

In addition to the requirement to "segment" the control buses to allow for partial failures, other enhancements were needed to the existing model. New malfunctions were needed to fail the RPCs and fuse. These malfunctions were required to produce the appropriate effects on the attached main buses rather than to be simple on/off failures. In addition, panels that were not failed by the instructor, but that were removed from any power source by other failed panels or malfunctions affecting the main bus supply power, had to be automatically failed. This last requirement proved to be the main design driver for the electrical power system model updates.

The SMS electrical power distribution system (EPDS) simulation model consists of two modules of Univac FORTRAN 11 code running in the SMS host computer, a Univac 1182 mainframe. One of these modules (the "logic module") calculates electrical power availability to the wide variety of orbiter buses. The other module (the "parameter module") calculates all bus voltages, currents, and loads. These two modules interface with each other, with the electrical power generation system simulation, with the instructor station, and with all SMS subsystem models requiring electrical power as an input and producing electrical load as an output. Due to the design option selected, the majority of model changes for this update were located in the logic module.

The general scheme followed in the EPDS model to determine power availability is to sum up all loads demanded from a bus and calculate the resulting bus voltage. If the voltage drops below a predetermined level, the power availability flag associated with the bus is set false. All models using the bus as an input monitor this flag. As described above, this scheme was already implemented for a "lumped" control bus. For the control buses, this scheme would be replaced with an algorithm that would calculate all the individual panel power availability statuses based on inputs from the instructor displays and power availability of the main buses feeding the control bus.

Due to the number of panel availability statuses that had to be calculated, an algorithm was sought that was efficient in terms of execution time. Figure 4 shows the control bus schematic annotated with the naming convention used in the algorithm that was finally developed. This naming convention will be briefly described. The RPC located closest to the voltage sensor is referred to as the "upstream RPC" and movement along the bus towards this RPC is spoken of as "in the upstream direction". Conversely, the other RPC is referred to as the "downstream RPC" and movement along the bus towards this RPC is spoken of as "in the downstream direction". The number of the panels along the bus is referred to as N and increases in the downstream direction. The most upstream panel is referred to as N=1. The most downstream panel is referred to as N_{LAST}. The panel to which the central main bus is connected by the fuse and circuit breaker is referred to as $N_{CENTRAL}$. The remainder of this paper will be devoted to an explication of the final algorithm. Figure 5 presents a flowchart of this algorithm.

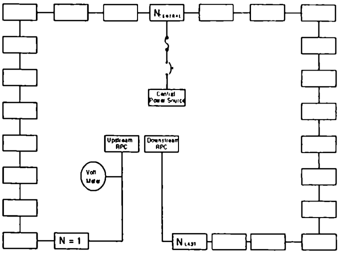


Figure 4. - Control Bus Model Naming Convention

Control bus processing is executed in a loop whose index ranges from 1 to 9, representing the nine control buses. At the beginning of each outer loop, an inner loop is executed to set all the panel statuses to "good". This was found necessary to allow for the recovery of power as failure scenarios unfolded. The "ACTIVATE/DEACTIVATE" flag is then checked. If active, all panels that have been malfunctioned from the displays have their statuses set to "failed". If any devices such as the RPCs or fuses have been malfunctioned, a model of their failure characteristics is executed here. This involves placing a larger than normal load on the main bus supplying power to the device for a typical failure duration. For the RPCs, a load sufficient to give a 7.5 ampere spike is placed on the bus for 1.5 seconds. For the fuses, a load sufficient to give a 30 ampere spike is placed on the bus for 200 milliseconds. After these times elapse, the status of the malfunctioned device is set to "failed".

After the malfunction processing is complete, a loop is executed whose index N ranges from 1 to N_{LAST}-1.

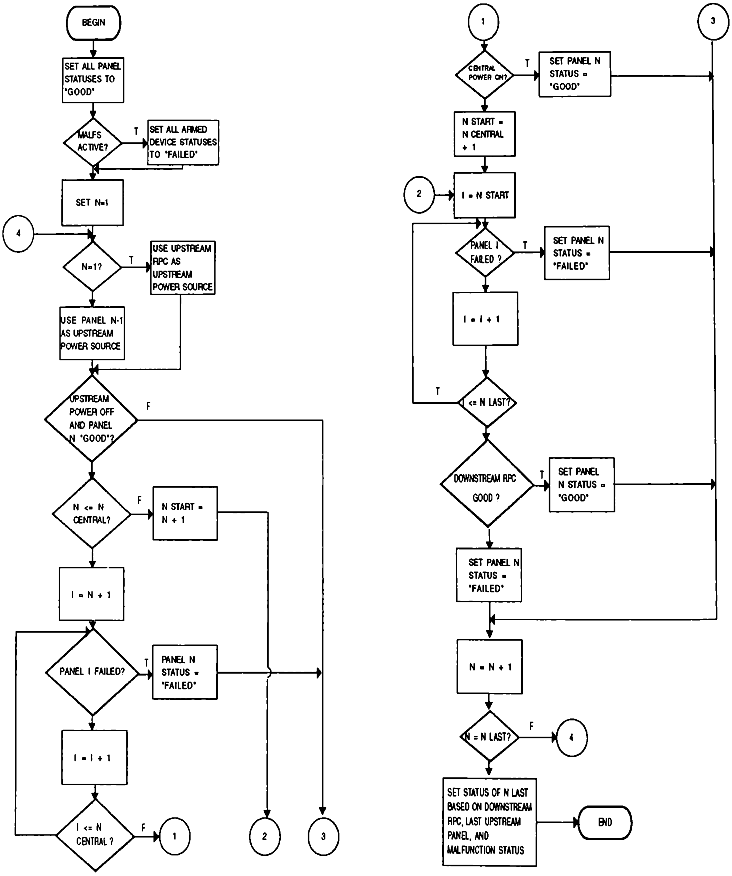

BEGIN
SET ALL PANEL STATUSES TO 'GOOD'
MALFS ACTIVE?
T
SET ALL ARMED DEVICE STATUSES TO 'FAILED'
SET N=1
4
N=1?
T
USE UPSTREAM RPC AS UPSTREAM POWER SOURCE
USE PANEL N-1 AS UPSTREAM POWER SOURCE
UPSTREAM POWER OFF AND PANEL N 'GOOD'?
F
N <= N CENTRAL?
F
N START = N + 1
I = N + 1
PANEL I FAILED?
T
PANEL N STATUS = 'FAILED'
I = I + 1
I <= N CENTRAL ?
F
1
2
3
1
3
CENTRAL POWER ON?
T
SET PANEL N STATUS = 'GOOD'
N START = N CENTRAL + 1
2
I = N START
PANEL I FAILED ?
T
SET PANEL N STATUS = 'FAILED'
I = I + 1
T
I <= N LAST?
DOWNSTREAM RPC GOOD ?
T
SET PANEL N STATUS = 'GOOD'
SET PANEL N STATUS = 'FAILED'
N = N + 1
N = N LAST?
F
4
SET STATUS OF N LAST BASED ON DOWNSTREAM RPC, LAST UPSTREAM PANEL, AND MALFUNCTION STATUS
END

Figure 5. - Control Bus Statusing Algorithm

First the panel being processed (panel N) is checked to see if it has been failed by the instructor. If it has, no further processing is performed for this panel and the loop counter is incremented. If the panel being processed is N=1, the upstream power source status is set to the upstream RPC status. Otherwise, the upstream power source status is set to the immediately upstream panel status.

If the upstream power source status is "good", or if panel N has been malfunctioned, no further processing is performed, since the current status of panel N is correct. If the upstream power source status is "failed", but panel N has not been malfunctioned, a search is performed in the downstream direction to determine if a path exists from panel N to a good power source without passing through any failed panels.

If panel N is downstream of $N_{CENTRAL}$, an inner loop is performed whose index I ranges from N+1 to $N_{CENTRAL}$. If any panel I is found to have a "failed" status, this means that panel N does not have a clear path to the central power source. Therefore the status of panel N is set to "failed" and no further processing is performed for panel N. When I becomes equal to $N_{CENTRAL}$, the status of the central power source is checked. If "good", the status of panel N is also set to "good", since this means that a clear path exists to a functional central power source.

If the central power source status is "failed", or if panel N is upstream of $N_{CENTRAL}$, another loop is executed. If this loop is entered due to the failed central power source, the loop index I will range from $N_{CENTRAL}+1$ to N_{LAST}. If entered due to panel N being upstream of $N_{CENTRAL}$, the loop index I will range from N to N_{LAST}. If any panel I is found to have a "failed" status, this means that panel N does not have a clear path to the downstream power source. Therefore the status of panel N is set to "failed" and no further processing is performed for panel N. When I becomes equal to N_{LAST}, the status of the downstream RPC is checked. If "good", the status of panel N is also set to "good", since this means that a clear path exists to a functional downstream RPC. If the downstream RPC status is "failed", the panel N status is set to "failed".

This completes the processing for a single panel. At this point the index N is incremented. If less than N_{LAST}-1, the processing is initiated for the next panel.

Special logic is required for panel number N_{LAST} due to the indexing schemes described above. Panel N_{LAST} has its status set explicitly. If panel N_{LAST} has been malfunctioned, or if neither the downstream RPC nor panel N_{LAST}-1 has a status of "good", the status of panel N_{LAST} is set to "failed", else it is set to "good".

The algorithm described has the advantage that for most scenarios, only the immediately upstream panel will be checked to determine a given panel status. This is possible due to the sequential nature of the algorithm which guarantees that the statuses of all panels upstream of the panel being tested have already been correctly set. The worst-case scenario would be one in which both the upstream and central power sources have failed. This means that the downstream RPC would be powering the entire control bus. The algorithm would then have to search downstream the entire length of the bus in order to status panel N=1 as "good". However, panels N=2 and higher would then be quickly statused due to the correct status of panel N=1. The downstream search does not have to be performed for each subsequent panel.

In order to train for more than one type of failure on a control bus, it was required that all existing control bus malfunctions be retained. Therefore, the portion of the old model that calculated the effects of the short malfunction was reused. The old-style fuse and RPC malfunctions were utilized in the new model where possible. This gives the instructor the capability to fail the entire control bus by using the old-style short malfunction. This malfunction, if entered at a sufficiently high value, will cause the fuse and RPCs to fail and therefore force the new algorithm to status all panels on the affected bus as "failed".

Concluding Remarks

The control bus model described in this paper was implemented in January 1991 and was first used for training crews and flight controllers for the STS-48 mission.

Since implementation of this model, additional testing has been done concerning the dangers of leaving a partially powered control bus feeding a short circuit. New flight rules and procedures are in place that would have the flight crew perform in-flight maintenance steps to totally disable the affected control bus. The new control bus model is being used to train crews and controllers on these procedures as well[6].

References

1. Tsiao, S. and Marena, N., "Introduction to Shuttle Mission Simulation," NASA/JSC Mission Operations Directorate, Training Division, JSC-19081, Aug. 1983.

2. "Space Shuttle Systems Handbook," NASA/JSC Mission Operations Directorate, Systems Division, JSC-11174, Dec. 1991.

3. Gregory, W. S., Neumann, W. R., and Smith, B. P., "Summary of the Control Bus Re-power Test," Lockheed Engineering and Management Services Company, LEMSCO-25658, May 1988.

4. "Space Shuttle Operational Flight Rules", NASA/JSC Mission Operations Directorate, JSC-12820, April 1992.

5. North, T. R., "SMS Control Bus Model Upgrade", Mission Operations Directorate, Training Division, SR005306, Nov. 1989.

6. Neely, J.C., "Shuttle Control Bus Kapton Wire Arc Tracking Test Summary", Lockheed Engineering and Management Services Company, EP5-M10-215, Oct. 1990.

AIAA-92-4144-CP

MODULAR DESIGN OF A MOVING-TARGET SIMULATION IN SUPPORT OF MISSION TRAINING OBJECTIVES

Mary D. Petryszyn, Avionics Systems Engineer
CAE-Link Corporation
Binghamton, New York

Abstract

Previous designs of moving-target simulation in flight simulators employed specific equations-of-motion (EOM) models to represent kinematics of specific moving-targets performing specific maneuvers. This implies that any changes to the type of moving-target or maneuver being simulated require changes to the equations-of-motion and/or software driving the maneuver. Also, the degree of fidelity required for the particular training maneuver being simulated is inherent to the software. This type of model is costly to maintain and update, and does not lend itself to efficient reusability. The B-2 Aircrew Training Device (ATD) program specification included the requirement to provide a moving-target simulation to be used in support of various mission training exercises, such as minimum-interval takeoff, aerial refueling, and navigation leg maneuvers. Moving-target simulation requirements also include the use of a variety of moving-targets, such as tanker, companion, and fighter aircraft, in support of these training exercises. Most recent programs, including the B-2 ATD, require the use of Ada as the standard programming language. These requirements, coupled with design goals of maintainability, reusability, and changeability, drove the development of a modular moving-target simulation subsystem that utilizes a generic equations-of-motion model.

Design Focus

The main focus in the design of any system must be to satisfy the requirements specified by the customer. This means developing a design that provides multiple moving-targets of various types active in the simulation at any given time. These moving-targets are all under independent control and each may be executing different types of maneuvers that require varying degrees of fidelity. For example, a tanker aircraft performing an aerial refueling maneuver requires the integration of wind effects on its motion, but a friendly aircraft performing navigation legs (point-to-point flight) does not.

Satisfying specified requirements also means developing the design utilizing an object-oriented design (OOD) approach and the Ada programming language. This OOD approach is different from the more traditionally applied functional approach to software development. Since the topic of this paper is not intended to be the concept of object-oriented design methodology, OOD will be described only in enough depth to understand both the definitions applied to the design of this system and terminology used throughout the paper.

Use of the Ada programming language means applying features such as efficient use of separate compilation units and parameter passing to aid in the modular development of the moving-target system design. By giving careful consideration to the structure of the system using these features, the design allows for easy correction of problems and can be readily enhanced.

In addition to satisfying requirements, another main consideration is the aspect of changeability. By identifying the potential areas of the system most likely to be susceptible to change throughout the life of the program, the design can be developed in such a way as to lend itself to easy, low impact updates. In the case of this moving-target design, the potential areas of change identified included:

a. Changes to the number of simulated moving-targets.

b. Adding new types of simulated moving-targets.

c. Adding new types of training exercises to be performed.

Finally, a great deal of consideration must be given to the issues of maintainability and reusability. This means developing a moving-target simulation design that includes a generic equations-of-motion (EOM) model that is adaptable to the specific characteristics of a given moving-target during real-time execution. This EOM model is then used to drive a specific moving-target to execute its associated maneuver to provide the required mission training objective. This makes the system adaptable to new applications and environments.

Background/Definitions

In the past, simulator software architectures employed functional designs. Moving-target systems were designed to simulate specific types of moving-targets performing specific maneuvers. These systems employed equations-of-motion models designed to represent the kinematics specific to the moving-target. The software was tailored to represent only the specific maneuver being simulated, with a specific degree of fidelity. Thus, when any change to the design was required, significant re-coding and compilation impacts were encountered. This type of design had a low maintainability factor, since the addition of a new type of moving-target or maneuver required the addition of equations similar to those already in place, which meant similar software had to be maintained in multiple areas. This type of design has a low reusability factor due to its specific nature.

In the OOD approach applied to develop the modular design described in this paper, an object is defined by the following characteristics:

a. An object represents a tangible or intangible real-world "thing".

b. An object consists of the specification of the data that represents the attributes/properties of the real world thing and the processes that operate on this data.

c. An object is an independent entity that has no (or low) coupling with other objects.

d. The object has a state -- it exists in time and space.

The attributes are the data which represent the object. The operations are the processes that manipulate the attributes. The terms "operation" and "procedure" are used interchangeably.

The term "motion profile" is used to refer to the specific automatic maneuver(s) associated with a particular moving-target. In this context, the moving-target motion profile relates to the training exercise to be performed.

The term "generic" with respect to the EOM model in this design is not to be confused with generic as it relates to the Ada programming language. It is intended to mean that the EOM model is adaptable to the specific performance characteristics of any given moving-target and to the fidelity required for the corresponding motion profile maneuver(s).

The term "import" will be used to refer to data brought into the moving-target system as input data from sources external to this system. Likewise, the term "export" will be used to refer to data output from this system for use by systems external to the moving-target system.

The automatic motion profiles define the role of each moving-target in the simulation in support of pre-planned mission training objectives. Instructor manual commands provide a way for these automatic profiles to be overridden during real-time, in order to allow a moving-target to be used in support of spontaneous training scenarios and to accommodate unplanned training objectives that develop during the course of mission execution and can be of great value to the student.

Development of Design Definition

An important task in the development of system object-oriented design is the object selection. In this design, the "moving-target" was selected as the object of the system. A moving-target represents a tangible, real-world entity that has definable properties and attributes and processes that operate on this data.

The next task is to define the object attributes and properties. A moving-target, in a sense, is an abstract entity in the world of simulation. Therefore, it is necessary to include attributes that may seem abstract to the definition of a real-world moving-target. Some of the moving-target object properties directly affect its motion and are used in its simulation; for example, its maximum velocity and acceleration. Other properties represent the object to other systems in the overall simulation, such as its visual description. Still other attributes may be used internally as well as externally, such as a moving-target's current latitude and longitude. The concept of how this system is connected to other systems in the overall flight simulation will be presented later.

For ease of discussion, the data representing the moving-target object can be placed into five main categories, as shown in Figure 1: 1) physical configuration, 2) motion performance characteristics, 3) automatic motion profile, 4) instructor commanded position, and 5) state.

The physical configuration defines what the moving-target "is". The following attributes fall into this category:

moving-target category (e.g., aircraft)
specific moving-target type (e.g., KC-135)
gross weight
landing gear status
speed brake configuration
lighting status
radio frequencies

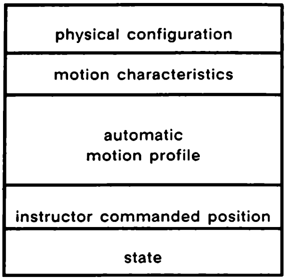


Figure 1. Moving-Target Object Definition

The motion performance characteristics define the physical limits of a moving-target relative to translational and rotational motion. This category of attributes includes the following:

maximum altitude
minimum/maximum speed
maximum acceleration/deceleration
maximum climb/dive rates
maximum bank angle
maximum bank rate
maximum bank acceleration
maximum pitch angle
rotate speed

The motion profile contains the information necessary to define a particular path of motion with respect to the earth. This data represents the moving-target's planned objective in the mission and is defined as follows:

motion profile type (e.g., navigation legs)
profile time data
initial position point latitude
initial position point longitude
initial position point altitude
initial position point speed
profile length (i.e., n = number of points
to move to)
profile position point 1 latitude
profile position point 1 longitude
profile position point 1 altitude
profile position point 1 speed
.
.
.
profile position point n latitude
profile position point n longitude
profile position point n altitude
profile position point n speed

The instructor commanded position defines the data necessary to manually override parameters of the planned mission profile of a moving-target. This data is defined as follows:

commanded heading
commanded altitude
commanded speed
commanded range
commanded relative bearing

The moving-target state defines the data necessary to describe the "condition of existence" of the moving-target at any point in time/space. This data includes the following:

control status (e.g., automatic)
profile status (i.e., what/where the moving-target is with respect to the profile execution)
ordered position (i.e., the specific point in space the moving-target is attempting to achieve)
current position (i.e., the specific point in space the moving-target currently holds)

After the object attributes are defined, it is necessary to define the processes that operate on these attributes. For this moving-target system, these procedures must consist of:

a. operations to determine the translational and rotational motion of a moving-target with respect to the earth (i.e., EOM's)

b. operations to drive the moving-target maneuver(s)

c. operations to implement the instructor commands

Design Detail

The procedures to execute each type of maneuver simulated are contained within separately compilable software units, as are the generic EOM's. These procedures are passed the attributes on which they must operate and they return the results of these operations via parameters. These resultant parameters may, in turn, be passed to other operations. With this type of software structure, procedures only have access to the data they need. This is facilitated by a software unit referred to as the system "controller" that acts as the single point of control within the system.

The motion profile data is used by the maneuver operations to automatically control the motion of a moving-target in support of specific mission training exercises. The maneuvers simulated include navigation legs, aerial refueling, and minimum-interval takeoff (MITO). The navigation leg motion profile represents simple point-to-point motion as defined by absolute position point data. The air refueling motion profile represents special aerial refueling paths, orbit, and emergency refueling maneuvers, and can also include taxi, takeoff, and fly-out motion. The MITO motion profile represents taxi, special MITO procedures, takeoff, fly-out, and navigation leg motion. The air refueling and MITO automatic motion profiles are more complex, representing motion paths that are responsive to the student in the simulation and are, therefore, defined by time and relative position data as well as absolute position point data.

Although the operations required to execute each of these motion profiles are different, the resultant output attributes are the same. The attributes passed to these procedures as input parameters consist of the respective automatic motion profile data defining the moving-target object. The output parameters returned consist of the ordered position state attributes.

It is possible to override the motion path specified by the automatic profile with instructor manual commands during real-time execution. The instructor commanded position attributes are the input parameters passed to operations that implement them and the ordered position state attributes are the returned output parameters.

The EOM model defines the operations that determine the translational and rotational motion of a moving-target with respect to the earth. The moving-target state ordered position and motion performance characteristics are passed to the operations as input parameters, while the current position state attributes are both passed to the operations as input parameters and returned from the operations as output parameters. The attributes of ordered position and current position are used by the EOM operations to drive the motion of the moving target. The motion performance characteristics are used by the generic EOM model during real time, to represent motion of a specific moving-target.

EOM Model Detailed Design

The equations-of-motion are driven by ordered heading, altitude, and speed values which are determined from the automatic motion profiles and manual commands. The equations use these values, along with performance characteristics specific to the type of moving-target being simulated, to compute current latitude, longitude, altitude, speed, heading, and attitude values with respect to the earth frame of reference.

Specifically, ordered values of altitude, speed, heading, bank, and pitch are passed through a second order filter (differential equation) to simulate moving-target response to a command. The ordered values represent the specific value in space that the moving-target must attempt to achieve, while the resultant values returned from the equations represent the current values in space that the moving-target holds for each given parameter.

The constants K_1 - K_5 which are passed as parameters and used by the EOM's are based on the Δt required for the moving-target. This allows a feature of adjustable rates in the EOM's as a function of target requirements. With this design, it is possible to update the position of the moving-targets at varying rates based on their role in the mission, or perhaps the fact that they are located in the visual scene which may require a faster update rate than those not in the visual scene.

Also included in this motion model is the element of a moving-target's specific performance characteristics. The values computed are limited by the moving-target performance characteristics. As an example, Figure 2 shows the EOM procedure and parameter passing scheme for altitude and the applied solution for altitude, altitude rate (i.e., rate of climb), and altitude acceleration.

The difference between ordered altitude and current altitude is computed as:

$$\Delta h = h_{ord} - h_c$$

The ordered value is passed through the differential equation:

$$h_c + 2\delta\omega \;\; h_c + \omega^2(\Delta h) = 0$$

In the general solution, the natural frequency, ω, was set to 1.0 and the damping ratio, δ, was set to 1.0. This equation is numerically solved to result in the following.

The altitude is computed as:

$$h_c = h_{ord} - \Delta h \;\; K_1 + (\dot{h}_c - \Delta h \;\; K_2) \;\; K_3$$

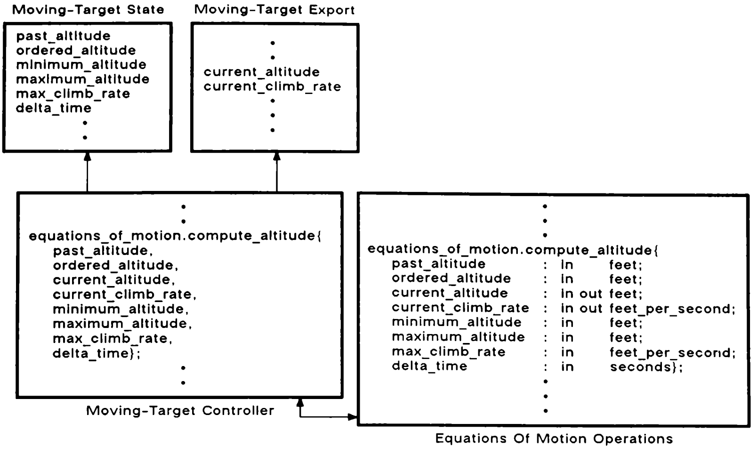


Figure 2. EOM Procedure Example

The altitude rate is then computed:

$$\dot{h}_c = \Delta h \;\; K_4 + (h_c - \Delta h \;\; K_2) \;\; K_5$$

Where:

$$\left.\begin{matrix} K_1 \\ K_2 \\ K_3 \\ K_4 \\ K_5 \end{matrix}\right\} \text{functions of time}$$

At this point, the moving-target altitude rate performance characteristic is applied to limit the computed value:

$$\dot{h}_{dive} < h_c < \dot{h}_{climb}$$

If it is necessary to limit the altitude rate value, then the altitude must be recomputed as:

$$h_c = (h_c)_{n-1} + \Delta t \;\; h_c$$

Next, the moving-target altitude performance characteristic is applied to limit the computed value:

$$h_{min} < h_c < h_{max}$$

Finally, the altitude acceleration is computed using the current and past altitude rate values:

$$h_c = [h_c - (h_c)_{n-1}]/\Delta t$$

The same type of computations as above are done for each of the motion parameters. Since the performance characteristics are passed to the equations-of-motion procedures as parameters, it is possible to alter them based on the type of profile the moving-target is executing. Perhaps the type of training being performed requires a moving-target to execute turns at half the standard rate. This is very easily accommodated by the design because no change is required to the actual equations; only the performance characteristic parameters must be altered. Also, this EOM model incorporates a degree of adjustable moving-target motion fidelity by allowing selective incorporation of parameters such as wind effects on the velocity vector and weight change effects on angle-of-attack.

With this structure, it is also very easy to implement instructor commanded heading, altitude, and speed values. This is done by first saving the ordered value of a commanded parameter for use later in returning to automatic control, and then setting the ordered value to the instructor commanded value of the corresponding parameter. Although this is transparent to the EOM's, this transitions the moving-target in this manual mode. When the instructor has finished with the moving-target in this manual mode, a command for return to automatic control is sent. This command is implemented by restoring the ordered parameter that was previously saved, thus putting the moving-target on a new course to its original destination.

Control of Moving-Target Simulation

The overall control of execution and data access within this system is managed from a separately compilable software unit. This "controller" is the only unit in the system that has direct visibility (i.e., access) to the import and export units and their data contents. The controller is also the only unit with access to the procedures contained in the motion maneuver and EOM units. The moving target system data flow is shown in Figure 3.

With this control structure, access to the data inputs required by any procedure is provided via parameters that are retrieved from imports and/or exports by the controller and passed into the procedures. Likewise, output parameters from a procedure are passed back to the controller and are then used to populate export data. This structure insures data integrity.

In this system, it is possible to have any number of moving-targets, from 0 to 15, active at any given time. The controller is responsible for managing what each of the currently active moving-targets is doing as well as when each one is processed. To make control decisions, this involves using state information for each moving-target, such as the type of motion profile being performed, whether the moving-target is currently under automatic or manual control or if it has completed its profile (action in the mission), and whether or not it is currently in the visual scene. The controller uses this information to determine which motion maneuver and EOM procedures must be invoked for each moving-target, how often a moving-target must be processed, and whether to identify this moving-target for replacement.

For example, the position of a moving-target that is in the visual scene must be updated at a faster rate than those not in the visual scene in order to prevent a "stepping" effect visually. Also, moving-targets present in the visual scene must move with higher fidelity, including both translational and rotational motion, while those not present in the visual scene require only translational motion. Or, a moving-target performing an aerial refueling motion profile must use 1/2 standard-rate turns. The design used here easily accommodates these variable fidelity requirements by controlling the execution rate of each moving-target independently and by passing the motion performance characteristics to the EOM's as parameters, after making any necessary adjustments.

As an example, Figure 4 illustrates a portion of control logic.

Using the logic of Figure 4, first consider moving-target #1 with a navigation leg motion profile that is

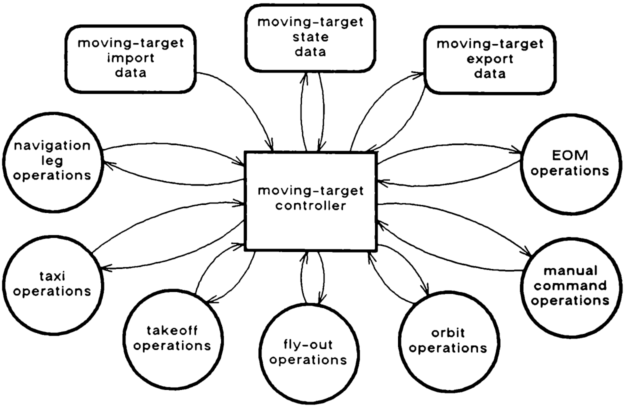


Figure 3. Moving-Target System Data Flow

```
If in visual scene or (not in visual scene and time to update) then

    determine $\Delta t$ as difference between last execution time and current time

    if motion profile = navigation leg then

        invoke procedure(s) to update navigation leg moving-target state

    elsif motion profile = air-refueling moving-target state

        invoke procedure(s) to update air-refueling moving-target state

    end if

    invoke EOM heading procedure
    invoke EOM altitude procedure
    invoke EOM velocity procedure
    invoke EOM latitude/longitude procedure

    if moving-target in visual scene then

        if motion profile = air-refueling then

                invoke EOM bank procedure with 1/2 standard rate turn characteristics

        else

                invoke EOM bank procedure with standard rate turn characteristics

        end if

        invoke EOM pitch procedure

    end if

end if
```

Figure 4. Control Logic Example

not located in the visual scene. It is easy to see that this moving-target will not be updated during each invocation of the moving-target controller, and when it is updated it will have a larger Δt (i.e., larger time difference between updates) than if it were in the visual scene. Only the maneuver operations necessary to update the state of a navigation leg moving-target will be invoked. Also, moving-target #1 will move only with translational motion since it is not in the visual scene.

On the other hand, consider moving-target #2 with an air-refueling motion profile that is located in the visual scene. This moving-target will be updated during each invocation of the moving-target controller and, therefore, will have a smaller Δt than moving-target #1; only the maneuver operations necessary to update the state of an air refueling moving-target will be invoked. In addition, moving-target #2 will be moved with both translational and rotational motion since it is in the visual scene. Finally, this moving-target will turn at 1/2 standard rate since it has an air-refueling motion profile.

The use of a control unit in this manner contributes greatly to the changeability goals of the design. For instance, if the required maximum number of active moving-targets were increased/decreased from 15, this number can be easily changed in one place and cause only minor compilation impact, since only the controller has visibility to, and dependence on, this number. No other code changes are necessary and no other units are affected.

This also holds true if a new type of motion profile were required. This change would require new code for the new maneuver procedures. Its addition to the current moving-target system, however, would cause only minor code changes and compilation impact to the controller to add access to the new procedures. Again, no other code changes are necessary and no other units are affected.

Finally, if the addition of a new type of moving-target were required, only the specific motion performance characteristics and physical configuration data must be provided. This should cause no code or compilation impact to the current moving-target system, as long as the data values fall within the currently specified range constraints on the data.

Connection to External Systems

The modularity of this design extends beyond the internal structure of the system to include the method by which it interfaces to other systems in the simulation. Within this system, the input data required from other systems during real-time execution is defined in a separately compilable import data software unit. The output data determined in this system is also declared in a separate export data unit.

The import data objects declared are populated by software external to this moving-target system during the simulation. This software connects the input object required by this system to the output object of the external system providing the data. The moving-target system is not required to know from where the information came. In the same fashion, the export data output from this system is picked up and used as necessary by any other systems in the simulation. This structure isolates the moving-target system from being coupled to the structure of other systems in the simulation.

The only thing that must be common to the systems providing and/or using data is the definition of the data type (i.e., integer, float, etc.). For instance, the "specific moving-target type" is used by the Image Generation System for visual representation. The definition of this data item must be the same for both systems in order to obtain consistent representation of this moving-target in the simulation. Another example is the radio frequency data that is used by the Communications System. The units and ranges specified for this data must agree between systems.

This consistency is accomplished by defining the data type to be used for a data object. This definition is placed in a unit that is external to all systems in the simulation. It is then available to be used by any systems that require a connection to a particular data object. A good example of this is the use of common type definitions for earth-related positional data. A unit containing the type definitions for latitude, longitude, altitude, a nautical mile, etc., would be used by all systems in the simulation requiring the use of data of this type. Thus, consistency is maintained between systems and across the overall simulation. An example of this data type definition scheme is shown in Figure 5.

The only other connection this system has with the overall aircraft simulation is between the moving-target controller and the simulation executive control. This is the connection for invoking execution of the moving-target system.

Figure 6 shows the overall moving-target system.

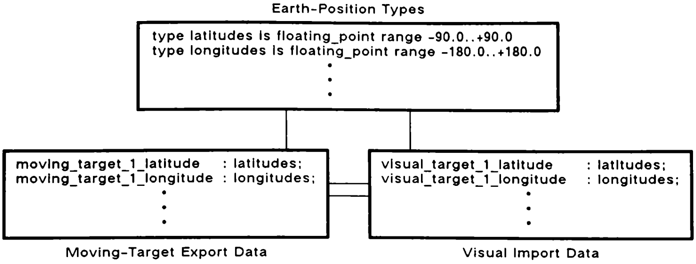

Earth-Position Types
type latitudes is floating_point range -90.0..+90.0
type longitudes is floating_point range -180.0..+180.0
moving_target_1_latitude : latitudes;
moving_target_1_longitude : longitudes;
visual_target_1_latitude : latitudes;
visual_target_1_longitude : longitudes;
Moving-Target Export Data
Visual Import Data

Figure 5. Moving-Target Data Type Example

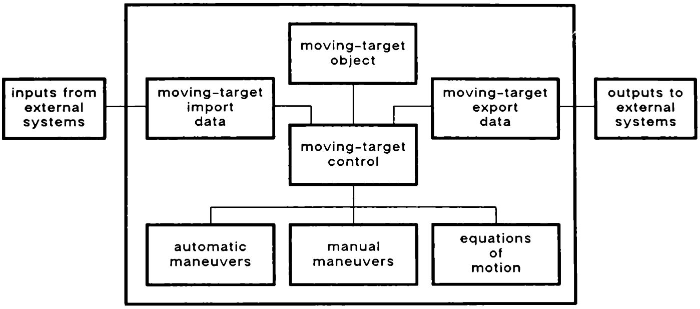

inputs from external systems
moving-target import data
moving-target object
moving-target control
moving-target export data
outputs to external systems
automatic maneuvers
manual maneuvers
equations of motion

Figure 6. Moving-Target System

Conclusion

The applied characteristics of the Ada programming language, such as parameter passing, and the practical application of object-oriented design (OOD) methodology contributed to and supported the modular design of this moving-target subsystem. Implementation of these concepts in structuring the software resulted in a subsystem that fulfills the specified requirements and is highly maintainable, easily changeable, and reusable.

The object-oriented structure used allows the moving-target system to be highly maintainable by loosely coupling the pieces of the system to permit low-impact changes. If the required number of moving-targets, type of moving-targets, or types of motion profiles simulated is expanded, the necessary changes can be made easily and efficiently due to isolated impact. This also relates to the reusability goals in that these important parameters in the design of this moving-target system are those that may most likely differ from one program to another. In addition, the execution and data control within this system, as well as the connection structure to external systems, contribute greatly to its potential reusability since the system is virtually self-contained. The methodology, structure, and programming language characteristics applied in the development of this modular moving-target design resulted in a system that not only satisfies the program requirements, but also meets the imposed design goals.

About the Author

Mary D. Petryszyn is an Avionics Systems Engineer with Link Flight Simulation Division of CAE-Link Corporation. She holds a B.S. in Electrical Engineering from Clarkson University and a M.S. in Computer Engineering from Syracuse University. Her present work includes systems engineering in the tactics simulation area for the B-2 Aircrew Training Device (ATD).

AIAA-92-4145-CP

A High-Fidelity Batch Simulation Environment for Integrated Batch and Piloted Air Combat Simulation Analysis

Kenneth H. Goodrich* and Dr. John W. McManus**
NASA Langley Research Center, Hampton Virginia

Alan R. Chappell†
Lockheed Engineering & Sciences Company, Hampton Virginia

Abstract

A batch air combat simulation environment known as the Tactical Maneuvering Simulator (TMS) is presented. The TMS serves as a tool for developing and evaluating tactical maneuvering logics. The environment can also be used to evaluate the tactical implications of perturbations to aircraft performance or supporting systems. The TMS is capable of simulating air combat between any number of engagement participants, with practical limits imposed by computer memory and processing power. Aircraft are modeled using equations of motion, control laws, aerodynamics and propulsive characteristics equivalent to those used in high-fidelity piloted simulation. Databases representative of a modern high-performance aircraft with and without thrust-vectoring capability are included. To simplify the task of developing and implementing maneuvering logics in the TMS, an outer-loop control system known as the Tactical Autopilot (TA) is implemented in the aircraft simulation model. The TA converts guidance commands issued by computerized maneuvering logics in the form of desired angle-of-attack and wind axis-bank angle into inputs to the inner-loop control augmentation system of the aircraft. This report describes the capabilities and operation of the TMS.

Introduction

As new technologies or capabilities are proposed for inclusion in high-performance aircraft, it is imperative to assess the impact, utilization, and costs of these technologies within the context of air combat tactics and effectiveness. Due to the highly complex and transient nature of air combat, simulation is the primary tool for performing this assessment. Both batch and real-time, piloted simulation can contribute to the assessment. Batch air combat simulations such as the Advanced Air-to-Air System Performance Model (AASPEM, Reference 1) and TAC BRAWLER (Reference 2) allow the study of aircraft tactics and performance in a highly controlled and repeatable environment. Batch air combat simulations consist of two fundamental elements--computerized maneuvering logics which generate maneuver decisions, and a simulation environment in which maneuvering logics are developed and tested. These programs can run a large number of engagements with minimal operator intervention, allowing comprehensive sets of initial conditions or parametric variations to be rapidly evaluated. Unfortunately, the minimal operator intervention inherent in batch operation slows development and validation of new maneuvering logics, resulting in a relatively inflexible set of tactics which may not effectively exploit a given situation or aircraft capability (Reference 3). In contrast, piloted simulation provides an environment ideally suited for rapid tactical experimentation and adaptation. New tactics can be investigated by simply instructing pilots to maneuver in the desired manner. Furthermore, the natural interface provided to the pilots encourages their participation in this development process and enhances their ability to assess the success of a given tactic. Unfortunately, due to the variability introduced by human pilots, the length of time required to perform a statistically meaningful piloted air combat simulation study, combined with the availability and expense of the necessary facilities and pilots, makes a comprehensive study extremely difficult to perform.

Because the strengths and weaknesses of batch and piloted simulation are complimentary, a synergism exists when the two approaches are employed in concert. To fully exploit this synergy, NASA Langley Research Center is developing an integrated batch and piloted simulation tool known as the Tactical Guidance Research and Evaluation System (TiGRES, Reference 3). TiGRES consists of three primary elements: an

* Research Engineer, Member AIAA

** Aerospace Engineer

† Senior Engineer, P.E., Member AIAA

advanced, real-time-capable, artificial intelligence-based maneuvering logic (Reference 4), a multi-dome, piloted simulation facility known as the Differential Maneuvering Simulator (DMS, Reference 5), and a batch simulation environment known as the Tactical Maneuvering Simulator (TMS). The development and operation of the TMS and its relation to the other elements of TiGRES are the focuses of this paper.

Unlike existing batch air combat simulation environments which typically use reduced order dynamic models, aircraft in the TMS are modeled using equations of motion, control laws, aerodynamics and propulsive characteristics identical to those used in high-fidelity piloted simulation in the DMS. This commonality allows maneuvering logics developed in the TMS to be evaluated, without modification, against human pilots in the DMS. The ability to test maneuvering logics against human pilots provides an efficient means of validating the results of batch simulation analysis. Thus, extensive preliminary investigations of tactical maneuvering strategies, guidance concepts or aircraft performance characteristics can be performed quickly and cheaply using the TMS. After the focus of an investigation matures, a minimum amount of piloted simulation in the DMS can be used to confirm or refine the findings of the more comprehensive batch analysis.

The TMS can be broken into three basic elements. The first element is the simulation model, used for simulating individual aircraft. Currently, models representative of a modern high-performance aircraft with and without thrust-vectoring capability are available. The second element is the Tactical Autopilot (TA). The TA is an autopilot which enables maneuvering logics to command full-order dynamic aircraft models in both the TMS and DMS. The TA converts guidance commands issued in the form of desired angle-of-attack and wind-axis bank angle into inputs to the inner-loop control augmentation system of the simulated aircraft. The final element is the TMS executive program and synchronization subroutine. This element provides the capability to simulate many vs many air combat by executing multiple, single aircraft simulations in parallel over a network of computer workstations. This paper will describe the three elements of the TMS and provide a demonstration of the operation of the simulation environment.

Description of the Tactical Maneuvering Simulator

Aircraft Simulation Model

Individual aircraft are modeled using a modified version of an existing batch simulation model developed at Langley Research Center. This simulation models a F-18 aircraft with or without a hypothetical, hardware-based thrust-vectoring (TV) system developed by the Northrop Corporation. This TV system uses two vectoring vanes on each engine to provide thrust induced pitch and yaw moments. When necessary to distinguish between the aircraft equipped with the TV system from the basic aircraft, the basic aircraft will be referred to as the baseline aircraft, while the aircraft with TV will be referred to as the TV aircraft. The batch simulation was developed from the real-time simulation code for the F-18 implemented in the DMS and from documentation obtained from the McDonnell Aircraft Company (ref 6-9). While an in-depth description of the batch simulation is to be published by its authors in a forthcoming NASA report, details relevant to its use in the TMS are presented here.

The computer code implementing the simulation model is written in the Advanced Continuous Simulation Language (ACSL, Reference 10) and FORTRAN. ACSL is a simulation system consisting of a special purpose high-level language, a translator, and various libraries to satisfy the commands available in the language. ACSL simulation models are translated into FORTRAN and linked with the ACSL libraries. The resulting executable program allows interactive user input, and enables the generation of plots and printed outputs. ACSL allows FORTRAN subroutines to be integrated into the simulation model. The simulation uses ACSL to implement the dynamics of the aircraft and engines. Actuator and sensor models are also implemented in ACSL. FORTRAN subroutines are used to calculate aerodynamic forces and moments and steady-state engine parameters. The discrete, inner-loop, control augmentation system of the aircraft is also implemented primarily in FORTRAN.

The equations of motion used in the simulation model the flight of a rigid airplane over a flat, nonrotating Earth using a conventional 6 degree-of freedom (d.o.f.) Euler formulation. The mass and moments of inertia of the aircraft are set at the start of a simulation and are assumed to be constant. Typical weights and moments of inertia used for the baseline and thrust-vectored aircraft are shown in Table 1. The configuration of the aerodynamic surfaces and controls is shown in Figure 1. The aerodynamic force and moment generated by each surface or control is calculated from a large wind-tunnel-derived database using table look-ups with linear interpolation. Data is stored in nondimensional form as functions of the air data variables [angle-of-attack(α), angle-of-sideslip (β), Mach number (M)], the time rate-of-change of α and β, surface deflections, and the body angular rates (p, q, r). The α range is [-10..+90] degrees, β range is [-20..+20] degrees, and M range is [0.2..2.0]. Flexibility effects in the form of flex/rigid ratios and flexibility increments are included in the database to an altitude of 60,000ft. Actuators for all control surfaces except the speedbrake are modeled with a first order lag with time constants and rate limiting, as appropriate. The actuator responsible for moving the speedbrake is modeled as producing a constant deflection rate of 24 degrees per second.

Two engines rated at 16100 lbs of installed static sea level thrust are included in the simulated aircraft. The engine model takes input from the throttle and current air data [altitude, dynamic pressure ($\bar{q}$), and M] to compute the force currently being produced by the engines. In the case of the TV aircraft, α and β effects as well as thrust losses due to vectoring are included in the thrust computation. The TV system consists of a two vane per engine installation as shown in Figure 2. By deflecting the thrust of the two engines in a symmetric or anti-symmetric manner, nearly pure pitching or yawing moments can be generated in a manner similar to an aerodynamic V-tail. The actuators for the thrust vectoring vanes are modeled as first order transfer functions with a steady-state gain of one, a time constant of (1/30) seconds, rate limits of 80 degrees per second, and position limits of [± 30°].

The simulated aircraft depends on full authority control augmentation system (CAS) to provide desirable flying qualities throughout its flight envelope. This CAS is documented for the baseline aircraft in detail in References 8 and 9. A simulation of the "Auto Flap Up" mode of the CAS defined by the 8.3.3 production PROM (programmable read only memory) set is included in the simulation model. The CAS used with TV aircraft is a refined and extended version of the baseline CAS. This work was performed by the Flight Dynamics Branch at NASA Langley through extensive batch and piloted simulation analysis. The CAS integrates the TV system with the aerodynamic control surfaces to significantly increase the maneuvering capabilities of the aircraft at high-α. The pitch and yaw commands from the command paths of the CAS are divided, as appropriate, between the aerodynamic and TV controls. The pitch and yaw commands sent to the TV system are passed through a mixer which resolves the commands into appropriate vane deflection commands for the thrust-vectoring hardware of the left and right engines.

The CAS augments the dynamics of the bare airframe to provide stability and predictable flying qualities which enable pilots to successfully employ the aircraft in tactical engagements. For use in the TMS, an outer-loop control system is needed around the basic CAS to perform the task of tracking trajectories as commanded by the TDG. In a sense, this outer-loop control system performs the physical functions of the pilot--transforming the desired tactical plan or strategy into actual aircraft motions. This outer-loop control system, known as the Tactical Autopilot, is described in the following section.

Tactical Autopilot

Most batch air combat simulation environments use simplified aircraft models which only model the steady-state characteristics of an aircraft. Frequently referred to as 5 d.o.f. models, these models are essentially point-mass representations with limitations on the rate at which the aircraft lift vector can be changed in magnitude and orientation. These limitations are selected to reflect the pitch, roll, and yaw capabilities of the simulated aircraft and usually take the form of a set of maximum allowable angular rates. The number of degrees of freedom is 5 rather than 6 because the aircraft is assumed to be coordinated at all times (defined as flight with β=0), requiring $r = p\tan\alpha$. Because no differential equations are used to describe the rotational dynamics of the aircraft, the aircraft orientation can be commanded directly, making the task of executing maneuvers specified by a maneuvering logic trivial.

In contrast, the TMS uses a full 6 d.o.f. representation of aircraft motion in which both forces and moments are used in the calculation of translational and rotational accelerations. This approach provides an accurate model of transient aircraft motions and is necessary to achieve commonality with piloted simulation models. The difficulty with using this higher-fidelity model is that aircraft attitude can no longer be commanded directly, requiring the addition of an outer-loop autopilot to execute maneuvers commanded by the maneuvering logic. Unlike traditional autopilots, this control system must be able to respond to the large amplitude commands typical of air combat in minimum or near minimum time. In the TMS, the TA (tactical autopilot) has been developed to perform this task.

The function of the TA is to accept trajectory commands generated by the TDG and issue commands to the inner-loop CAS which cause the aircraft to follow the desired trajectory. The TDG issues trajectory commands by specifying a desired α and wind axis bank angle (μ), defined as

$$\mu=\tan^{-1}\left[\frac{\sin\theta\cos\alpha\sin\beta+\sin\phi\cos\theta\cos\beta-\cos\phi\cos\theta\sin\alpha\sin\beta}{\sin\theta\sin\alpha + \cos\phi\cos\theta\cos\alpha}\right],$$

combined with a desired throttle and speedbrake setting. Flight with β=0 is assumed to be desired at all times.

For a given flight condition, these parameters determine the magnitude and orientation of the net force vector acting on the aircraft and the attitude or the aircraft relative to an Earth fixed reference system. Since the throttle and speedbrake settings can be obtained directly, no interface is needed to capture these commands; the commands are passed directly from the TDG to the aircraft simulation. The TA thus serves as an all-attitude, outer-loop control system to capture and track α and μ as commanded by the TDG. A block diagram of the complete TDG-TA-aircraft system is shown in Figure 3. It should be recognized that while the TA is described in this report in the context of the TMS, its use is also required in the DMS. By incorporating the TA into the piloted simulation model used in the DMS, the TDG is able to command this simulation in a manner identical to the batch simulation. The design and development of the TA is described in detail in Reference 11 and will only be briefly described in this report.

The TA is divided into two channels; a longitudinal command system which uses longitudinal stick inputs to capture and track commanded α, and a lateral command system which uses lateral stick inputs to capture and track the commanded μ. A directional controller is not included in the TA because the inner-loop CAS already attempts to maintain zero sideslip, unless commanded not to by using rudder pedal inputs. Piloted simulations have shown that the wind-axis rolling performance of the baseline aircraft can be improved slightly at α's greater than 25° using rudder pedal inputs (Reference 12). This behavior is not being exploited by the current implementation of the TA.

The longitudinal command system uses a proportional-integral-derivative (PID) structure with α feedback. The lateral command system uses a proportional-derivative (PD) structure with μ feedback.

The values of α, $\dot{\alpha}$, μ, and $\dot{\mu}$ are assumed to be available without error, thus no additional compensation to account for sensor noise or dynamics is included in the TA. There is no attempt to model in the TA the cognitive and neuromuscular delays or limitations that would be inherent in a human pilot. Thus, as implemented, the TA represents an idealized controller.

One of the difficulties in developing a system such as the TA is determining suitable criteria to measure the acceptability of the final design. Traditional performance specifications such as frequency and damping are not appropriate considering the large-amplitude, coupled maneuvers performed by the TA. Criteria which reflect the nonlinearities of the task must be used to assess the performance of the TA. The intent of these criteria is to insure that the TA is able to capture and track commands from the TDG in a manner which does not adversely bias the tactical performance of the TDG-TA-aircraft system. Since this tactical performance is dependent on the combined interactions of all three elements, it is desirable to characterize the response of the TA-aircraft system against some functional benchmark. Since the only previous controllers to demonstrate mastery of the simulated aircraft in air combat maneuvering are human pilots, the performance of pilots performing representative maneuvers should provide a reasonable benchmark for the performance of the TA.

Tables 2 and 3 show the minimum and average time required for a series of experienced pilots to perform large amplitude, decoupled α and μ captures in the baseline and TV aircraft, as simulated in the DMS. Also shown in the tables is the time required by the TA to perform the same captures. All runs start from one-g, level flight and end when the desired α or μ is captured within the specified tolerance. The tables show that for all but two of the tasks, the TA required less time than the minimum time used by the pilots. The TA is probably able to consistently perform the desired maneuvers in less time than the human pilots due to its ability to respond instantly to the current situation. In the two tasks in which the TA did not outperform the pilots, the performance differences are small.

For the 90° roll maneuver at α=10° with the TV aircraft, the TA takes 0.06 seconds longer than the minimum piloted time. This increase is probably tactically insignificant and may be due to α variations during the maneuver. Data recorded during the execution of the maneuver show that the pilot allowed α to fall to 7.2° during the maneuver; the TA experienced a minimum α of 8.5°.

For the 40° α capture task at Mach 0.6 with the baseline aircraft, the TA was unable to prevent the initial overshoot from exceeding the desired ±2.0° capture tolerance. This overshoot increased the capture time of the TA for the original capture tolerance beyond the minimum piloted time. The initial overshoot experienced by the TA was 0.44° beyond the desired capture tolerance. As this overshoot only slightly exceeds the desired capture tolerance, the tactical performance should not be significantly affected. Since attempts to improve the response at this one condition resulted in an overall decrease in system performance, the decision was made to accept nominal response of the system. The time listed in Table 3 represents the performance of the TA with the capture criteria relaxed to 2.44°.

Also shown in the tables is the maximum peak overshoot (M_p) for the captures performed by the TA. Burgin, in Reference 13, recommends that for good tactical performance, Mp for decoupled inputs be limited to 5° in pitch and 20° in roll, regardless of the amplitude of the input. For all the captures, the TA is below these recommended limits.

The capture tasks shown above measure performance for single-axis, step-inputs. In air combat, the TA will be expected to respond to sequences of simultaneous α and μ commands. The response of the TA to a representative command sequence is shown in Figures 4 and 5 for the baseline and TV aircraft, respectively. These command sequences were obtained by discretizing, at one second intervals, continuous α and μ time histories recorded during piloted ACM engagements. This discretization was performed to obtain command sequences representative of the command update rate of the TDG. Because the sequences were obtained from actual trajectories, they should be reasonably close to the capabilities of the TA controlled aircraft and representative of a tactically realistic sequence.

The TA appears to follow both sequences with sufficient accuracy to effectively implement realistic maneuver sequences. As shown in Figures 4 and 5, the ability of the TA to capture and maintain α and μ is only slightly reduced by the coupled command sequences. It should be noted that an absolute, operational assessment of the effectiveness of the TA cannot be performed until the system is interfaced with an appropriate TDG and tested against human pilots in the DMS.

Multiple Aircraft Simulation

In contrast to most batch simulation environments which are implemented as a single large process, the TMS uses a concurrent implementation structure to provide multi-aircraft simulation. This parallel implementation allows a single copy of the simulation program to be run concurrently as needed to simulate the individual engagement participants. The number of concurrent copies of the simulation which can be executed simultaneously is only limited by available computer memory and the desired execution speed--of course, an appropriate TDG would be needed to command this number of aircraft.

Parallel implementation offers several other key advantages over conventional methods. Since all aircraft are simulated by the same program, corrections or updates to this model need only be performed once, thus easing configuration control issues. The parallel implementation also allows different simulation models to be incorporated into the TMS and intermixed with the current aircraft simulation model with the addition of a standard subroutine. Thus, simulations of different aircraft types can easily be added to the TMS, allowing comparisons of the tactical performance of dissimilar aircraft. Simulations which may be added to the TMS are not restricted to aircraft; high-fidelity missile simulations could also be implemented in a similar fashion. Finally, parallel implementation allows individual simulations to be distributed on multiple, networked computers, reducing the time required to simulate a given engagement.

While the concurrent parallel implementation provides the above mentioned benefits, it is necessary to provide a control mechanism to synchronize the independently executing simulations. This synchronization is required so that the simulations remain together on the same time step. Since the simulations execute as independent processes on a given computer (or set of computers), the order and length of time in which the computer operates on each process is a function of other jobs which may be executing on the machine and is essentially indeterminate. Thus, without some control mechanism, the simulations may progress at different rates.

The TMS utilizes a read-write synchronization protocol to suspend execution of individual processes at a specified point until all relevant processes have reached this point. The protocol is used in the TMS to suspend execution of the aircraft simulations at the end of the current time step or simulation frame. The simulations are allowed to proceed only after all the participating simulations have reached the end of the current time step and have received updated maneuvering commands from their controlling TDG.

The key elements of the parallel implementation are an executive program, a communication and synchronization subroutine called by the aircraft simulation model, and a specialized message-passing protocol. The executive program serves as a master process which initializes the individual simulation models and supervises their operation in a common reference frame. The executive program also manages communications with the TDG. Since all communication between a TDG and its corresponding aircraft must pass through the executive, the flow of information can be closely monitored and controlled. The communication and synchronization subroutine is called by the aircraft simulation model at the completion of each simulation frame. As the simulation interface to the message-passing protocol, this subroutine allows the executive program to suspend execution of the simulations, pass current state information from the simulations to the maneuvering logics, and return updated maneuver commands to the simulations at the end of the decision process.

The following section demonstrates the capabilities of the TMS with a sample engagement.

Demonstration of The Tactical Maneuvering Simulator

This example engagement demonstrates a one-versus-one dogfight between a drone aircraft following a predefined, open-loop command sequence and an aircraft actively guided by a simple TDG. The objective of this example is to demonstrate the operation of the TMS with an active TDG.

The TDG commands α and μ in an effort to cause the flight path of the guided aircraft to intersect a predicted future position of the drone aircraft. This predicted future position is obtained by extrapolating along a second order curve fit to the past three recorded positions of the drone aircraft. The TDG then determines the maneuver plane and load factor required to intercept this position given the current state of the guided aircraft. The maneuver plane is defined as the plane determined by the current velocity vector and the net force vector acting on the aircraft. The required maneuver plane and load factor are converted into a required α and μ. If the required load factor is outside the aerodynamic or structural capabilities of the aircraft, α corresponding to maximum available or allowable lift is commanded. In addition, if the commanded μ differs from the current μ by more than 45° and the commanded α is greater than 15°, the α command is reduced to 15° in order to expedite the execution of the rolling maneuver. This α reduction was heuristically selected and does not necessarily reflect an optimum maneuvering strategy.

The engagement between the two aircraft is shown in Figure 6 from various perspectives. The engagement starts with both aircraft trimmed in 1g level flight at an altitude of 10,000 ft and Mach=0.9. The aircraft start from opposite headings with a 10,000 ft longitudinal separation and a 1,000 ft lateral offset. The drone aircraft is initially commanded to maintain μ=0° and increase α slightly over the trim value. The throttle of the drone aircraft is advanced into the afterburner region. These commands are maintained during the first 10 seconds of the engagement. After the initial merge, the guided aircraft performs an oblique, pitch-back maneuver to reverse its heading back toward the drone aircraft. Following this initial 10 second period, the drone is commanded to increase α to 28° and alternate μ between ±90°, switching every 10 seconds. The resulting motion is a descending spiral like trajectory. In response to these maneuvers, the guided aircraft reverses its heading again, and effectively tracks the drone down the descending spiral. Time histories comparing commanded α to actual α and commanded μ to actual μ for the guided aircraft are shown in Figure 7. These time histories demonstrate that the TA controlled aircraft is able to closely track the guidance commands generated by the TDG.

Current Research Activities

With the basic development of the environment completed, the TMS, as part of TiGRES, is being used to investigate and develop tactics for highly agile aircraft. The tactical capability of the TV equipped aircraft is to be compared with the baseline aircraft in 1 versus 1 and 1 versus 2 scenarios. This comparison requires the development of a TDG capable of maneuvering the aircraft effectively in these scenarios. A prototype TDG known as the Computerized Logic for Air Warfare Simulation (CLAWS) has been developed for 1 versus 1 air combat using simplified 5 dof aircraft models (Reference 4). An extension of CLAWS, known as Paladin, has been interfaced with the TMS and is currently being evaluated with the high-fidelity aircraft models used in the TMS (Reference 14).

The TA as described in this paper only supports guidance commands in the form of a desired α and μ. These parameters are useful for commanding the trajectory of an aircraft during the gross maneuvering phases of air combat maneuvering. However, when a target has been acquired, and fine tracking is required to achieve a weapons solution, having a direct means of "aiming" the aircraft is desirable. To support this desire, a second mode of operation is being added to the TA. This mode will allow the TDG to designate a target to the TA and the TA will then use a conventional feedback control law to minimize the line-of-sight error to the target.

Concluding Remarks

The development and operation of a batch air combat simulation environment known as the Tactical Maneuvering Simulator has been presented. The Tactical Maneuvering Simulator serves as a tool for developing and evaluating tactical maneuvering logics. The environment can also be used to evaluate the tactical implications of perturbations to aircraft performance and supporting systems.

The Tactical Maneuvering Simulator was developed using an existing batch simulation of a modern high-performance aircraft, with and without thrust-vectoring. This batch simulation uses 6 degree-of-freedom equations of motion, aerodynamics, propulsive characteristics, and control laws equivalent to those used in high-fidelity piloted simulation.

An outer-loop control system known as the Tactical Autopilot was developed to allow maneuvering logics to command the 6 degree-of-freedom aircraft model. The Tactical Autopilot uses longitudinal and lateral stick inputs to capture angle-of-attack and wind-axis bank angle as commanded by the maneuvering logic. The performance of the Tactical Autopilot was demonstrated by comparing the time required for it to capture decoupled angle-of-attack and bank-angle commands to the time required by human pilots for the same commands. The performance of the the Tactical Autopilot was equivalent or superior to the pilots for nearly all the commands investigated. The ability of the Tactical Autopilot to track realistic command sequences of angle-of-attack and bank-angle was demonstrated using sequences generated from piloted air combat simulations. The Tactical Autopilot was shown to effectively track these representative command sequences.

To provide for the simulation of air combat with multiple participants, a parallel implementation scheme was developed using a read-write synchronization protocol. This parallel implementation allows the Tactical Maneuvering Simulator to simulate air combat with any number of engagement participants. The maximum number of participants is limited only by the available computer resources. The parallel implementation is also beneficial from the standpoint of simplifying software maintenance and allowing new simulations to be easily added to the environment.

The capabilities of the Tactical Maneuvering Simulator were demonstrated with an example engagement. This engagement demonstrated the ability of the environment to simulate multiple aircraft and to interact with an active tactical decision generator. The tactical autopilot was shown to closely follow the maneuver commands from the tactical decision generator.

References

1. Advanced Air-to-Air System Performance Model (ASSPEM) Users Manual, Boeing Document D180-28938-1, November 1985.

2. Kerchner, R. M.; et al.: The TAC BRAWLER Air Combat Simulation Analyst Manual (Rev. 3.0). Decision Science Applications Report #668, 1985.

3. Goodrich, K. H.; and McManus, J. W.: Development of a Tactical Guidance Research and Evaluation System (TiGRES). AIAA Paper 89-3312, August 1989.

4. McManus, J. W.; and Goodrich, K. H.: Application of Artificial Intelligence (AI) Programming Techniques To Tactical Guidance for Fighter Aircraft. AIAA Paper 89-3525, August 1989.

5. Ashworth, B. R.; and Kahlbaum, William M., Jr: Description and Performance of the Langley Differential Maneuvering Simulator. NASA TN D-7304, 1973.

6. F/A-18 Stability and Control Data Report, Vol I: Low Angle of Attack. McDonnell Aircraft Company, MDC A7247, November 1982.

7. F/A-18 Stability and Control Data Report, Vol II: High Angle of Attack. McDonnell Aircraft Company, MDC A7247, August 1981.

8. F/A-18 Flight Control Electronic Set Control Laws, Volume I. McDonnell Aircraft Company, MDC A4107, July 1988.

9. F/A-18A Flight Control System Design Report, Volume I, System Description and Theory of Operation. McDonnell Aircraft Company, MDC A7813, September 1984.

10. *Advanced Continuous Simulation Language (ACSL) Reference Manual, Fourth Edition.* Mitchell and Gauthier Associates, 1986.

11. Goodrich, K.H.: Development of a Six Degree-of-Freedom Simulation Environment for Tactical Guidance Research and Evaluation. Masters Thesis, George Washington University, April 1991.

12. Murphy, P. C.; Bailey, M. L.; Ostroff, A. J.: Candidate Control Design Metrics for an Agile Fighter. NASA TM-4238, March 1991.

13. Burgin, G. H.; Eggleston D. M.: Design of an All-Attitude Flight Control System to Execute Commanded Bank Angles and Angles of Attack. NASA CR-145004, February 1976.

14. Chappell, A. R.; McManus, J. W.; Goodrich, K. H.: Trail Maneuver Generation and Selection In the Paladin Tactical Decision Generation System. AIAA Paper 92-4541, August 1992.

Table 1 - Summary of Weight, CG and Inertias

		CG Locations		Moments and Products of Inertia			
	Weight	Fuselage Station	Water Line	I_{xx}	I_{yy}	I_{zz}	I_{xz}
	(lbs)	(in)		(slug - ft^2)			
Thrust-Vectored	33310	455.0	102.8	23000	151293	169945	-2971
Baseline	31665	457.3	101.6	22337	120293	138945	-2430

Table 2 - Time Required by TA to Perform α Captures

All runs started at altitude=25,000ft

Aircraft	Initial α (deg)	Final α (deg)	Initial Mach	Capture Criteria	Average Time by Pilot (sec)	Minimum Time by Pilot (sec)	Time by TA (sec)	Maximum Overshoot (deg)
Baseline	4.4	30.0	0.60	±2°	5.12	4.35	**1.91**	**1.9**
"	4.4	40.0	0.60	"	2.88	2.30	**2.28***	**2.4**
"	23.5	30.0	0.30	"	4.93	3.78	**1.00**	**1.4**
"	23.5	40.0	0.30	"	6.56	5.95	**1.81**	**1.6**
"	10.0	0.0	0.40	"	2.50	1.99	**1.34**	**1.0**
"	20.0	0.0	0.32	"	5.86	5.25	**1.88**	**2.0**
"	30.0	0.0	0.27	"	7.06	5.68	**2.38**	**2.0**
TV	4.4	30.0	0.60	"	4.70	3.84	**1.09**	**1.7**
"	4.4	40.0	0.60	"	4.45	3.46	**2.97**	**2.6**
"	4.4	50.0	0.60	"	4.76	5.31	**2.41**	**0.2**
"	23.5	30.0	0.30	"	2.11	1.09	**0.81**	**1.2**
"	23.5	40.0	0.30	"	2.69	1.41	**1.38**	**1.2**
"	23.5	50.0	0.30	"	3.39	1.79	**1.78**	**1.6**
"	10.0	0.0	0.40	"	2.18	2.18	**1.12**	**0.4**
"	20.0	0.0	0.32	"	2.11	1.66	**1.60**	**0.7**
"	30.0	0.0	0.27	"	4.60	4.54	**1.89**	**0.6**

* Capture criteria relaxed to 2.4°

Table 3 - Time Required by TA to Perform 90° μ Captures

All runs started at altitude=25,000ft

Aircraft	Initial α (deg)	Initial μ (deg)	Final μ (deg)	Capture Criteria	Average Time by Pilot (sec)	Minimum Time by Pilot (sec)	Time by TA (sec)	Maximum Overshoot (deg)
Baseline	10.0	0.0	90.0	± 5°	4.10	3.07	**1.43**	**3.8**
"	20.0	0.0	90.0	± 8°	8.90	6.70	**4.90**	**6.00**
TV	10.0	0.0	90.0	± 5°	2.15	1.47	**1.53**	**2.8**
"	20.0	0.0	90.0	"	5.00	4.40	**2.22**	**2.7**
"	30.0	0.0	90.0	"	5.17	2.75	**2.50**	**3.9**

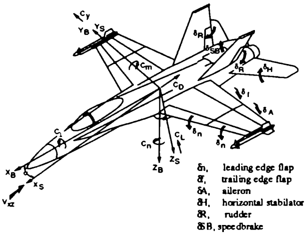
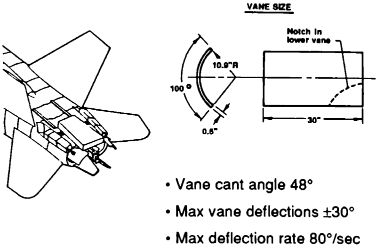


Figure 1 - Configuration of Aerodynamic Surfaces
and Definition of Axes



- Vane cant angle 48°
- Max vane deflections ±30°
- Max deflection rate 80°/sec

Figure 2 - Thrust-vectoring system

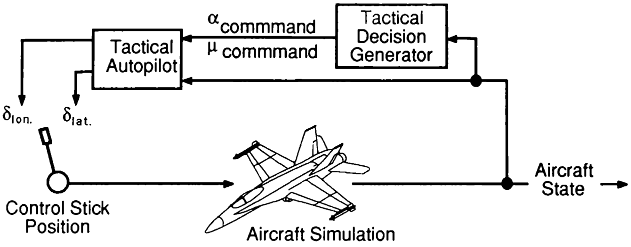


Figure 3 - Block diagram of TDG - TA - Aircraft system

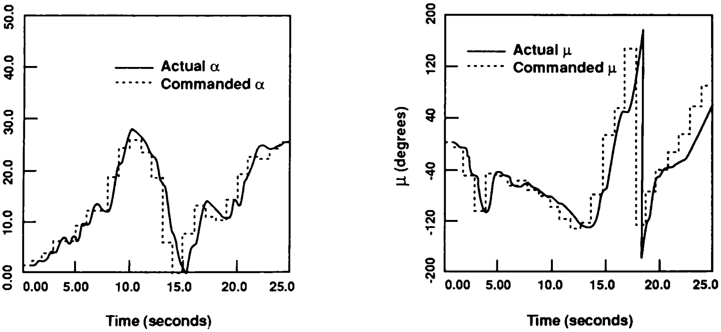


Figure 4 - Response of TA to command sequence
Baseline aircraft

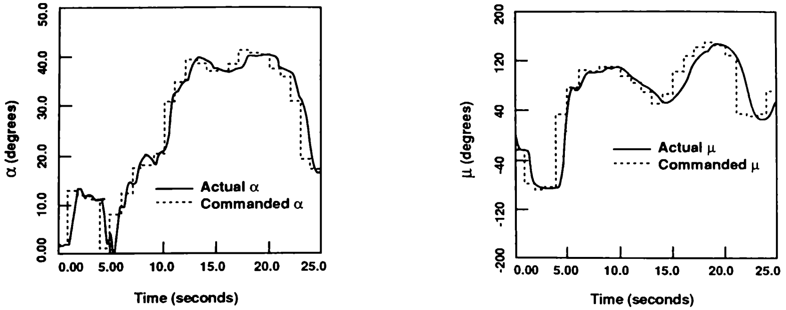


Figure 5 - Response of TA to command sequence
TV aircraft

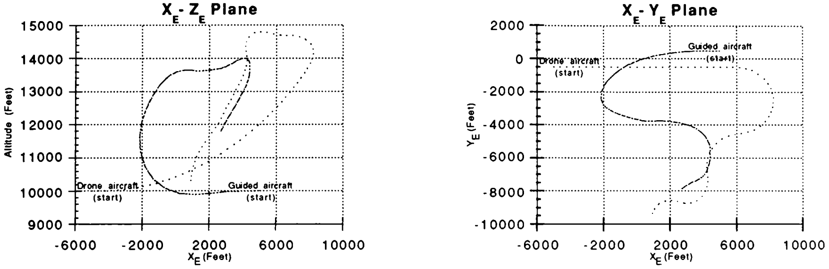


Figure 6 - Example 1 versus 1 engagement

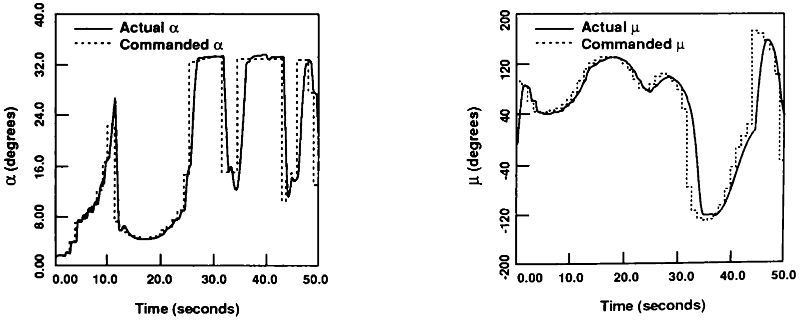


Figure 7 - Response of TA to command sequence from TDG
Baseline aircraft

AIAA-92-4147-CP

AIAA Paper No. 92-4147, AIAA Flight Simulation Technologies Conference
24-26 August 1992, Hilton Head, South Carolina

REDUCING TRANSPORT DELAY THROUGH IMPROVEMENTS IN REAL-TIME PROGRAM FLOW

by

R. Marshall Smith*
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665-5225

ABSTRACT

This paper will describe the process of measuring and reducing software transport delays through careful analysis, and modifications of real-time programs. A 737 program is analyzed and modified to improve the simulation overall transport delay by approximately 30%. The transport delay was improved through modification to the real-time program flow and the implementation of quaternions in the calculation of the math model. This study resulted from a measurement study implemented and reported on last year. The measurement methods and equipment used is described in more detail in that report.

INTRODUCTION

Transport delay is defined as the amount of time elapsed, in addition to intended math model delays, from a pilot input until an appropriate response is made to the pilot by the associated simulation hardware and software. The transport delay can be divided into hardware transport delay and software transport delay. Hardware transport delay includes the delay associated with particular piece(s) of hardware and non-changing software. For example, the Computer Generated Image (CGI) system contains real-time software and databases that do not change on a regular basis. Since this delay remains constant from one vehicle model to another it is quantified as a single transport delay. Hardware transport delay is typically the largest contributor to the overall transport delay. For example, some CGI systems require an average of 110 ms to process and display the received data. Hardware transport delay can include delays due to sampling, computer iteration rates. Software transport delay is defined as the delay associated solely with the vehicle model and real-time program structure that is implemented. Therefore, the hardware transport delay is any delay resulting from a piece of simulation equipment that is not associated with the math model. Since there may be several hardware paths a signal may take back to the pilot (Motion base, CGI, instrumentation, etc.) from his control input, there may be several hardware transport delays associated with a particular simulation.

The adverse effects of excessive transport delay have been well documented.[1,2] Excessive delays can lead to degraded pilot performance and, if the transport delays are too long, simulator sickness can occur from mismatched cueing[2]. Therefore, it is critical that the transport delays associated with a particular simulator be quantified and reduced, when necessary, to an acceptable range. The Federal Aviation Administration (FAA) and independent testing have confirmed that it is desirable to keep transport delays under 150 milliseconds (ms) to avoid degraded pilot performance[3]; however, there is evidence that longer delay times may be tolerated for vehicles that have lower response rates (such as transport aircraft)[1].

During the past year, the transport delays for the simulators in the Flight Simulation Facility (FSF) at NASA Langley Research Center (LaRC) were measured and documented[4]. The primary function of the FSF is to conduct basic aerospace research. In conducting this research daily changes are required to be made to many different math models. Also, each of these research simulations may require the use of different pieces of equipment. The FSF uses

*Engineer, member AIAA

a centralized approach to simulation. Resources common to multiple simulators are shared to improve productivity. The FSF is capable of supporting four simultaneous simulations. Each simulation begins with the real-time computers. The FSF has two Control Data CYBER 175 real-time computers. However, these computers are currently being replaced by two Convex computers, a Convex C2 and a C3[5]. This study was conducted using the CYBER 175. From the CYBER a real-time job can be scheduled and executed. The simulation engineer who is responsible for initializing the program schedules the required simulation hardware through the Advanced Real-Time Simulation system (ARTS)[6]. The engineer can select from eight flight decks; two CGI systems; three graphics subsystems, called the Calligraphic/Raster Display System (CRDS); two CGI target generators, for dome simulations; and various other subsystems a particular simulation might require. Therefore, depending on the configuration selected, a particular simulation will usually have multiple signal paths.

This study will document the transport delays associated with the Boeing 737 software executed on the Transport Systems Research Vehicle (TSRV) simulator. In addition, methods to reduce the software transport delay will be introduced. The transport delays were measured in the time domain using a logic analyzer. Data collection was performed using a step input at the cockpit and a Video Level Detector (VLD) at the monitor. The VLD detects video levels at the input of the monitor and outputs to the logic analyzer a trigger signal to indicate an event has occurred. Reference 4 contains the details on how the measurement method was selected.

SIMULATION REQUIREMENTS

Figure 1 shows the flight simulation hardware used. The 737 simulation executes on a CYBER 175 and interfaces to the required simulation equipment through the ARTS system. The ARTS system uses the CAMAC crates to interface hardware to the fiber optic highway. Each crate contains all the necessary Analog-to-Digital Converters (ADCs), Digital-to-Analog Converters (DACs) and digital data interfaces for a particular site. The 737 simulation requires the CGI, (an Evans & Sutherland CT6) for out-the-window visuals, one third of the CRDS, (a Terabit Eagle 1000), to generate eight Heads-Down Displays (HDD), and the TSRV cockpit. The cockpit is composed of dual McFadden sidearm control loaders, discrete switches and two Lear Siegler Cockpit Display Units (CDU)[7]. The CYBER 175 executes the simulation at 33 Hz (or once every 30 ms). The CGI updates four, 771 line displays at 50 Hz (or once every 20 ms). The update rate for the CRDS is a function of the complexity of the graphical image being drawn. The more complex the image the slower the update rate. The update rate for the baseline TSRV displays is 40 Hz. The update rate for the displays used in this study is 60 Hz. Due to the different update rates throughout the system there are several points of asynchronous communication. Reference 4 discusses the asynchronous delays present in the FSF. These nodes can be modeled as a Zero-Order-Hold (ZOH) device with the average transport delay of one half the update rate[8].

TEST SETUP

Hardware Setup

The simulation hardware is setup as described in reference 4. Figure 2 shows the signal paths taken by the 737 simulation. Since all measurements are conducted in the time domain a step function is injected onto the ARTS highway in place of the output of the control loader. The step function is digitized and read into the CYBER for processing. Depending upon the math model implemented, the CYBER calculates the next state information for the simulation and outputs that information to the appropriate device (CGI, CRDS, et. cetera). The CGI and the CRDS then generate their output based upon the latest input.

Software Setup

The ADC Sample, ARTS/CYBER delay is measured from the input of a step at the ADC, node 3, to the output of the system, received at the CGI crate and/or the Eagle crate, nodes 4 and 6. The input ADC converts the position output from the control loader into digital data suitable for the CYBER. The data is read into the CYBER at the beginning of each real-time frame (every 30 ms). For testing without the math model, a statement was inserted into the real-time program that will cause the program to skip all model dependent executable statements and then output to the CGI and CRDS a signal that indicates the input ADC has detected a change at the cockpit. For testing with the math model, the aircraft's pitch was monitored for any deviation from a trimmed condition to signal a response from the input. First the aircraft is placed into a trimmed condition to stabilize the pitch variable, THETADEG, or θ_{deg}. Once θ_{deg} has changed by a preset threshold amount the appropriate output is made to the CGI and CRDS indicating that the input has traveled through the math model. This is not to say that the output of the math model is correct, that requires more detailed analysis, but only that the onset of a response to a pitch input is occurring[4].

INITIAL RESULTS

Hardware Results

The transport delay for each section of the signal path was measured. The hardware transport delay is measured with the math model bypassed. Table 1 shows the individual hardware transport delays for each section of the path. The TSRV's total (average) hardware transport delay time was 129.9 milliseconds (ms) to the end of the first CGI field, 131.5 ms to the end of a 40 Hz CRDS display, and 94.7 ms to the end of a 60 Hz CRDS display[4]. A timeline of these delays can be found in figure 3.

Software Results

The math model was introduced into the computation flow and the test was repeated. It can be seen in Table 1 that the math model added approximately 63 ms to the overall transport delay for all hardware paths. The math model was expected to only add approximately 30 ms to the total transport delay. This additional delay not only can be detrimental to the simulation but also exceeds the 150 ms specified by the FAA. To correct this problem an investigation was initiated to determine why the output was delayed an extra 33 ms. A timeline of these delays can be found in figure 4.

CORRECTIVE MEASURES

Modify Simulation Program Flow

The first place to investigate to reduce the transport delay is to chart the computational flow of the real-time program. To obtain the lowest transport delay, the simulation program should be outputting any data that requires further computing (CGI, CRDS, et. cetera) as soon as possible. To determine when data was leaving the real-time program, a print statement was inserted into the TSRV code to determine what values were being output during which CYBER frame. From this method, it was determined that the data was changing sometime during the third frame. Ideally, the output data should change in the same frame that the input was received. Therefore, since the data should be ready to be sent at the end of the first frame, a study of the program flow was implemented.

TSRV Program Structure

For the purposes of this paper, a simplified view of the program structure of the TSRV is presented in figure 5. The major subroutine calls that calculate the next state of the math model and output the data to the hardware data are shown in the order they are called. The path that an input in pitch would take is described in figure 6. The input arrives at the beginning of the frame and is first processed in ACCEL to generate QBDOT, which is the pitch acceleration in the body axis. QBDOT is then integrated at the end of the frame, in IGRATE6, to yield QB. The integration method used is the Adams Bashforth (AB-2) method, then program is recycled at the end of the frame[12]. THETADOT, the velocity in the earth axis, is then generated in DERIV and integrated to yield THETA at the end of the frame. THETA is then converted from radians to degrees (THETADEG) in MISCEL. The data for the CGI and CRDS is output as soon as the calls to CGIINTF and CDCAGT are made, sometime in the third frame. This accounts for a minimum software delay of at least two frames.

The methodology behind this program flow is due in part to the integration method used. The integrations are located at the end of the frame because the AB-2 method is a prediction method and it will determine the value of the state variable at the frame time, t, plus the step size of the integration, h. Therefore, since the output of IGRATE6 is the state information for the beginning of the next frame it should be output near the beginning of the frame. This program structure has been in use since the early seventies and, at the time of its development, the only external simulation equipment was analog. In addition, data is only transmitted at the end of a frame. The analog systems typically had very low transport delays, therefore there was not much concern over whether the total transport delay was excessive. However, with the addition of digital systems to the network, such as the CGI and CRDS, the transport delay for the system has increased dramatically as well as the capability to transmit anytime in the frame.

Proposed TSRV Program Structure

From the description above, it is clear that the simulation program can be reordered to calculate and output time critical data earlier in the frame. Figure 7 shows the reordered subroutines. The object is to calculate new data as soon as possible, therefore, new control inputs are read in by TCVCAB, SWITCH, MSPLOG and INPUT at the beginning of the frame. The data is then used in ENGFM, LNGFM, AEROFM and SUMFM to calculate new forces and moments on the model. ACCEL and DERIV calculate the state variables and derivatives. Next a new subroutine, called QUAT is added. QUAT performs a transformation using the quaternion (or Euler parameter) rate equations on the accelerations from this frame and the velocities from the last frame. It also integrates these equations using the AB-2 method to yield positional information in the earth axis. Previously the transformation from the body to earth axis was done separately in the conversion from QDOT to THETADOT. This simultaneous integration and conversion saves one frame over the AB-2 method since positional information is generated in the first pass. IGRATE6 is then used to generate the velocities for this frame from the accelerations. This is because the output of QUAT only contains positional data. The conversion from radians to degrees is executed in the QUAT subroutine. RUNWAY converts positional coordinates from latitude and longitude to CGI database coordinates. The data is then output to the CGI. Finally, the remaining miscellaneous equations are calculated in MISCEL. The data is then sent to the CRDS and the cockpit through CDCAGT and DACS. The new pitch path is shown in figure 8.

These changes should accelerate the availability of data to the CGI and CRDS by almost two frames (60 ms). A speedup of exactly two frames is not possible since the calls, sending data to the CGI and CRDS, are made later in the frame. The time to output to the cockpit will be accelerated by exactly two frames since this data is shipped synchronously at the end of the frame.

RESULTS

Software Transport Delays

Once the changes described above were incorporated into the 737 real-time program, the output variables to the CGI and CRDS were printed again and observed to change within one frame. The total TSRV transport delay was retested using the same procedures, and the result is shown in Table 2. The transport delay resulting from the math model was 5.4 ms. Therefore, the total transport delay to the end of the first CGI field was measured at 135.0 ms. Because the call to send data to the CRDS was moved with the CGI call, and the output data was verified to change during the first frame, the transport delay through the CRDS was not remeasured. The new transport delay can be recalculated using the previous values for the CRDS update rates. The transport delay to the beginning of a 40 Hz display is 136.9 ms and 100.1 ms to the beginning of a 60 Hz display. A timeline of these delays can be found in figure 9.

Verifying Numerical Accuracy

To verify the numerical accuracy of the real-time program after the modifications were completed, duplicate variables were added to the program to allow the old program flows to be calculated (but not output). This allowed both versions of the 737 program to be compared while using the same real-time data. This method allows the direct comparison of any variable to determine if the program flows are numerically the same. After a detailed analysis was conducted no discrepancies in variable data were discovered. This is as expected since all the equations are still being executed in the same order as in the previous program flow. The only difference is that the data is now output in the frame it is calculated.

CONCLUSIONS

The original intent of this study was to develop a method of detecting and documenting transport delays within the FSF at NASA Langley Research Center. However, as the project grew, the data revealed several problem areas in the FSF that required immediate attention. It was discovered that the transport delay within the CGI was approximately 20 ms longer than expected. The manufacturer is currently modifying the CGI software to reduce the delay by approximately 17 ms[4]

A second problem was located in the software implementation of the program and in the program flow. The transport delay of the math model software was measured and found to be excessive. A detailed study of the real-time program implementation was conducted. The results of the study showed that the integration method used required two frames to calculate the next state data for the CGI and the CRDS. Then, although the data was ready, it was not output until sometime in the third frame. Through reordering of the 737 code and the addition of quaternion transformations to the program, the problem was corrected. This improved the TSRV transport delay by approximately 57 ms. Since these savings do not impact the computational accuracy of the math model, all the real-time programs at the FSF will be modified to reflect this program order.

With the unexpected delays eliminated from the CGI, and the software delays corrected, the TSRV will be accelerated by approximately 74 ms. Therefore the overall transport delay for the TSRV, including the math model, to the end of the first CGI field is reduced to approximately 118 ms (see Table 2). The transport delay to the end of a CRDS field is reduced to 137 ms for a 40 Hz display and 100 ms for a 60 Hz display. This lowers the overall transport delays, for all signal paths, to within the FAA guidelines of 150.0 ms.

The transport delays associated with other hardware configurations and math models were also measured. Many of the other programs used, at the FSF, already have quaternions implemented and only required the reordering of program flow. When these programs were reordered the math model transport delay was reduced by approximately 30 ms. Adding in the CGI transport delay improvement yields a total transport delay reduction, for these programs, of approximately 47 ms. It should be noted that overall transport delay was reduced without the addition of predictive algorithms to the code.

Future improvements can be made in reducing the transport delay by procuring faster hardware and running the simulations at higher iteration rates. NASA LaRC is currently procuring an Advanced CGI that will further reduce the CGI transport delay by 27 ms. In addition, all real-time programs will begin operation, on the Convex computers, at 60 Hz, beginning in August 1992. This will save at least 8 to 10 ms on all signal paths.

REFERENCES

1. Bailey, R.E., Knotts, L.H., Horowitz, S.J., & Malone, H.L. (1987, August). Effect of Time Delay on Manual Flight Control and Flying Qualities During In-Flight and Ground-Based Simulation. AIAA paper 87-2370, Monterey, CA

2. Hettinger, L.J., Lane, N.E., & Kennedy, R.S. (1988 September). Diagnostic Measurement Approaches to the Problem of Simulator Sickness in Flight Simulation. AIAA paper 88-4624-CP, Atlanta, GA

3. Federal Aviation Administration (1991 July 29). Airplane Simulator Qualification. Advisory Circular AC 120-40B

4. Smith, R., Marshall (1991 August). A Method for Determining Transport Delays in the Flight Simulation Environment. AIAA paper 91-2964, New Orleans, LA

5. Cleveland; J., & Sudik, S., & Grove, R. (1992 August). High Performance Flight Simulation at NASA Langley. AIAA paper 92-4179, Hilton Head, SC

6. Crawford, D., Cleveland, J., & Staib, R. (1988 August). The Langley Advanced Real-Time Simulation (ARTS) System. AIAA paper 88-4595, Atlanta, GA

7. NASA Langley Research Center (1990). Langley Aerospace Test Highlights. NASA Technical Memorandum 104090

8. Johnson, W.V., & Middendorf, M.S. (1988 September). Simulator Transport Delay Measurement Using Steady States Techniques. AIAA paper 88-4619-CP, Atlanta GA

TABLE 1 - Total TSRV Transport Delays (Average)

Device			No Math Model (ms)		With Math Model (ms)	
ADC Sample			15.0		15.0	
ARTS/CYBER			4.8		4.8	
Math Model			0.0		62.8	
CGI Sample			10.0		10.0	
CGI			100.1		100.1	
Total			129.9		192.7	
ADC Sample			15.0		15.0	
ARTS/CYBER			4.8		4.8	
Math Model			0.0		62.8	
CRDS Sample	(40)	(60)	12.5	8.3	12.5	8.3
CRDS	(40)	(60)	99.2	66.6	99.2	66.6
Total	(40)	(60)	131.5	94.7	194.3	157.5

TABLE 2 - Reordered TSRV Transport Delays (Average)

Device			With Math Model (ms)	
ADC Sample			15.0	
ARTS/CYBER			4.8	
Math Model			5.4	
CGI Sample			10.0	
CGI	(present)	(upgraded)	99.8	82.8
Total	(present)	(upgraded)	135.0	118.0
ADC Sample			15.0	
ARTS/CYBER			4.8	
Math Model			5.4	
CRDS Sample	(40)	(60)	12.5	8.3
CRDS	(40)	(60)	99.2	66.6
Total	(40)	(60)	136.9	100.1

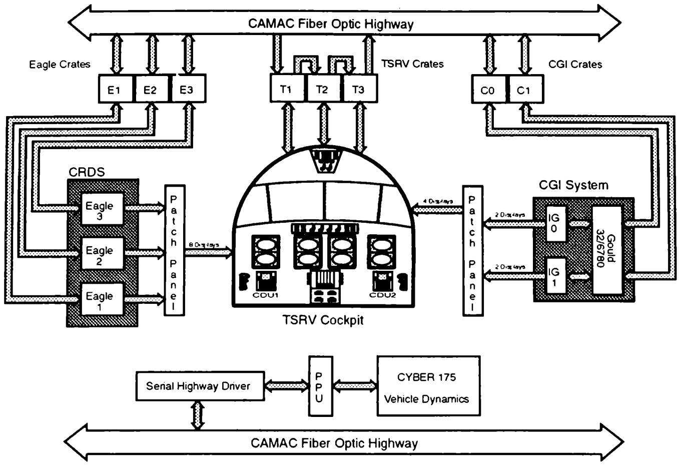


Figure 1 - TSRV Flight Simulation Hardware

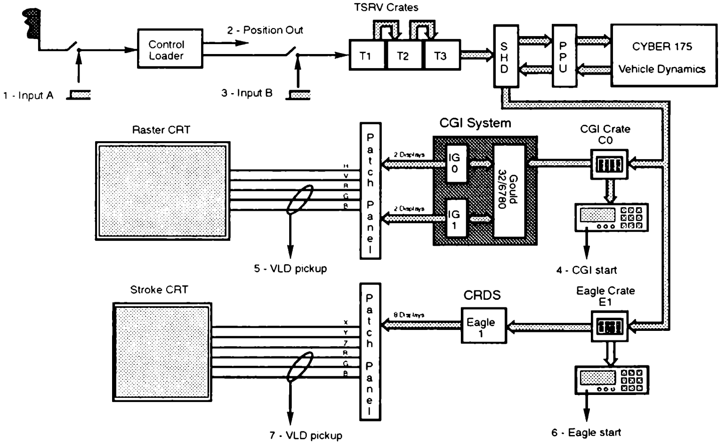


Figure 2 - Signal Flow Diagram

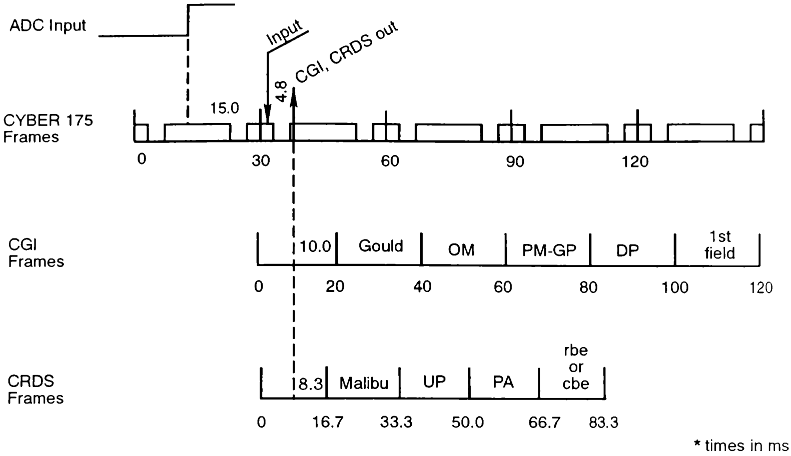


Figure 3 - TSRV Transport Delays (no math model)

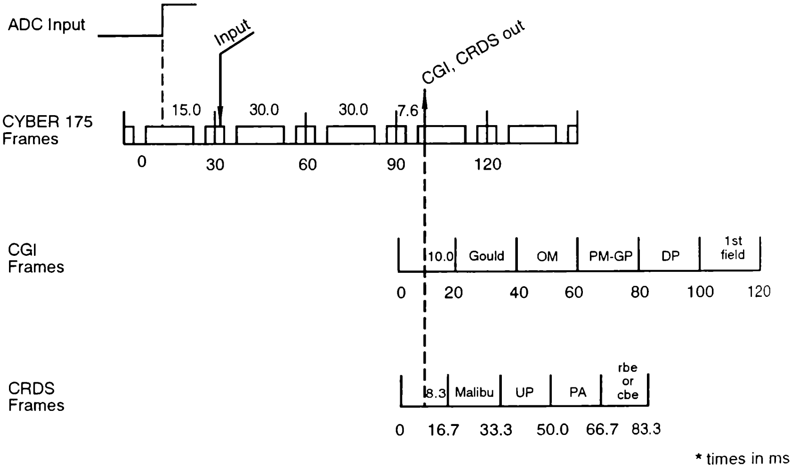


Figure 4 - TSRV Transport Delays (737 math model)

TCVCAB	-	read analog inputs from the cockpit
MISCEL	-	determine psi, theta, phi for the last frame, solve miscellaneous equations
RUNWAY	-	converts the lat and long to CGI database coordinates
ACCEL	-	compute new body accelerations (UDOT, VDOT, WDOT)
SWITCH	-	stores guidance and navigation data into buffers
MSPLOG	-	determines the flight mode selected from cockpit switches
INPUT	-	calls autopilot, SAS, engine model, and yaw damper
ENGFM	-	compute new engine forces and moments
LNGFM	-	compute new landing gear forces and moments
AEROFM	-	compute aerodynamic forces and moments
SUMFM	-	sumation of forces and moments
DERIV	-	calculate new derivatives and PDOT, QDOT, RDOT
CGIINTF	-	send old x, y, z, psi, theta, phi to the CGI
CDCAGT	-	send old data to the CRDS
DACS	-	send data to the analog controls in the cockpit
IGRATE6	-	compute new state variables and U, V, W, P, Q, R, using AB-2

Figure 5 - TSRV Program Flow

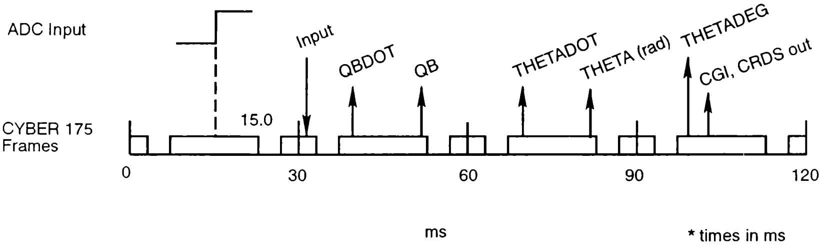


Figure 6 - TSRV Program Signal Path

TCVCAB	-	read analog inputs from the cockpit
SWITCH	-	stores guidance and navigation data into buffers
MSPLOG	-	determines the flight mode selected from cockpit switches
INPUT	-	calls autopilot, SAS, engine model, and yaw damper
ENGFM	-	compute new engine forces and moments
LNGFM	-	compute new landing gear forces and moments
AEROFM	-	compute aerodynamic forces and moments
SUMFM	-	sumation of forces and moments
ACCEL	-	compute new body accelerations (UDOT, VDOT, WDOT)
DERIV	-	calculate new derivatives and PDOT, QDOT, RDOT
QUAT	-	calculate new quaternion coefficients and psi, theta, phi
IGRATE6	-	compute new state variables and U, V, W, P, Q, R, using AB-2
RUNWAY	-	convert the lat and long to the CGI database coordinates
CGIINTF	-	send new x, y, z, psi, theta, phi to the CGI
MISCEL	-	solve all miscellaneous equations
CDCAGT	-	send new data to the CRDS
DACS	-	send new data to the analog controls in the cockpit

Figure 7 - Reordered TSRV Program Flow

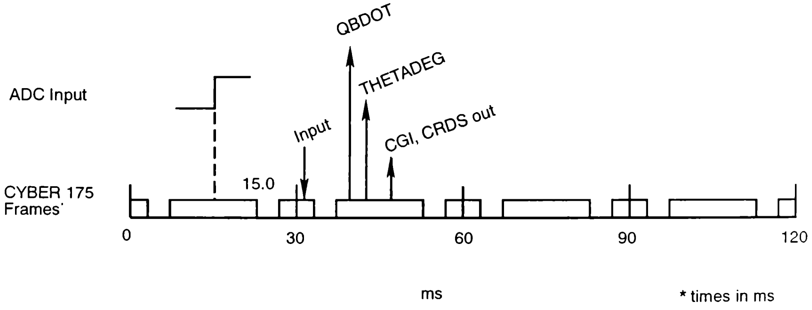


Figure 8 - New TSRV Program Signal Path

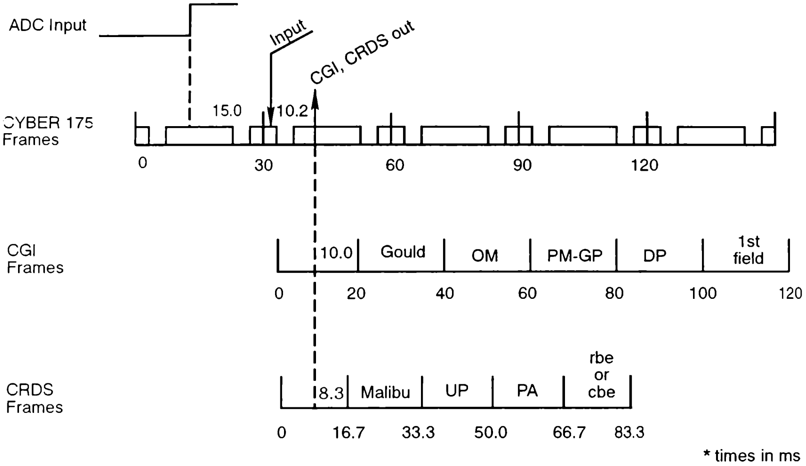


Figure 9 - Reordered TSRV Transport Delays

AIAA-92-4149-CP

DEVELOPMENT OF A REAL-TIME SIMULATION OF A SHIP-CORRELATED AIRWAKE MODEL INTERFACED WITH A ROTORCRAFT DYNAMIC MODEL*

Warren F. Clement**
Peter J. Gorder
Wayne F. Jewell**
Systems Technology, Inc.
2672 Bayshore Parkway, Suite 505
Mountain View, CA 94043
(415) 961-4674

ABSTRACT

This paper describes a study outlining approaches for modeling mathematically in real-time simulation the kinematics of a ship and the dynamics of a rotorcraft interacting aerodynamically within the ship's airwake on approach and landing aboard the moving ship. Specific recommendations are made to modify an extant real-time simulation model of a rotorcraft, including the interfacing of the ship-correlated airwake model with the blade element aerodynamic model of the rotor. A method for modeling the three-dimensional, spatially correlated airwake is also presented.

NOMENCLATURE

A_1, B_1	MX*NY matrices containing the coefficients of the MX sine and cosine terms, respectively, in the NY sums of sinusoids
B	A dimensionless numerical constant on the order of ten
$\underline{c}_0^{\prime}, C_0$	An NY row vector containing the NY mean values for each sum of sinusoids
$\underline{c}_1^{\prime}, C_1$	An NY row vector containing the NY trend values for each sum of sinusoids
DI	Dynamic interface
$E(\Omega)$	Turbulent energy spectral density
$F_{u_1}(j\omega)$	Describing function of one-dimensional (1D) point correlated spectral filter
f	Abbreviation of f_{pu_1} in Table 3
f_{pu_1}	Overall power fraction (dmls)
$G_{u_1k}(j\omega, \Delta\eta)$	Describing function of two-dimensional (2D) laterally correlated spectral filter
GENHEL	Generic helicopter
GTOSL	Gust table offset laterally
$H_{u_1k}(j\omega)$	Describing function of the sum of the independent random and mutually correlated fluctuating velocities
h	Perturbed height displacement with respect to inertial space (ft) (+ up)
i	Summation index
j	$\sqrt{-1}$; also an indicial subscript
$K_0(\alpha\eta)$	Modified Bessel function of the second kind, order zero
$K_1(\alpha\eta)$	Modified Bessel function of the second kind, order one
k, TKE	Turbulent kinetic energy (units are L^2/T^2)
LHA	General purpose amphibious assault ship (Landing Helicopter Assault)
MX	Integer representing the number of complex coefficients retained in the computation of the finite Fourier transform (FFT) of a vector of length $NX (MX \leq NX/2)$
N_B	Number of main rotor blades
N_{SE}	Number of rotor blade segments per blade
NX	Integer representing the dimension of the X-coordinate of a data base
n	Independent Gaussian white noise source
nD	n-dimensional
PHOENICS	Parabolic Hyperbolic Or Elliptic Numerical Integration Code Series
PSD	Power spectral density
pdB	Power decibels (10 $\log_{10}$ of the power ratio)
R_{MR}	Main rotor radius (ft)
R_T	Rotor radius in general (ft)
$T_{S/I}$	Transformation from inertial to ship axes
TPOSL	Turbulence parameter offset laterally
$\bar{U}_{w/I}$	Ship's longitudinal component of local mean wind velocity with respect to inertial space at designated stationkeeping location
u_1	Longitudinal gust velocity component (units are L/T)
u_2	Lateral gust velocity component (units are L/T)
u_3	Normal gust velocity component (units are L/T)
$\overline{u^{\prime 2}}$	Mean-squared value of the locally isotropic fluctuating velocity component $u^{\prime}$ (units are L/T), $\overline{u^{\prime 2}} = 2k/3$
$V_{FLD/S}$	Gust propagation velocity collinear with wind over deck (WOD) in GENHEL (Ref. 11, ¶5.9.2) (units are L/T)
V_{KT}	True airspeed of the rotorcraft (kt)
$V_{OFLD/S}$	Component of the rotorcraft velocity relative to inertial space and orthogonal to the wind velocity over the deck
$\underline{V}_{S/I}$	Velocity of the ship with respect to inertial space
$\underline{V}_{W/I}$	Velocity of the wind with respect inertial space
$\underline{V}_{W/S}$	Velocity of the wind with respect to the ship
X_{TPT}	Width of the turbulence parameter tables
x	longitudinal displacement; perturbed longitudinal displacement with respect to inertial space (ft)
y	Lateral displacement; perturbed lateral displacement with respect to inertial space (ft)
z	Vertical displacement; perturbed vertical displacement with respect to inertial space
α	Defined to be $\sqrt{\Omega_e^2 + \Omega_1^2}$
ϵ, epsilon	Kinetic energy dissipation rate (units are L^2/T^3)
η	Lateral or spanwise displacement (in the horizontal plane) orthogonal to the direction of the mean wind or air mass velocity (units are L)
ξ	Normalized offset of rotor blade hinge (dmls); $\xi = e/R_T$
π	3.14159...
$\overset{(1)}{\Phi}_{xx}(\Omega_1)$	One-dimensional power spectral density of x
σ	Root-mean-squared (RMS) value of a variable specified by a subscript
$\overset{(2)}{\Phi}_{xx}(\Omega_1, \eta)$	Laterally correlated two-dimensional power spectral density of x
ψ_{RWOD}	Relative wind direction over the deck
Ω_1	Spatial wave number coordinate in the direction of the gust propagation velocity (units are 1/L), $\Omega_1 = (\omega_{e1})/V_{FLD}$
Ω_e	Spatial wave number associated with the energy-containing eddies in the airwake of the ship (units 1/L) $\Omega_e = (A\epsilon)/(\overline{u^{\prime 3}})$, where A is on the order of unity

*Work sponsored in part by the Naval Air Systems Command under Contract N00019-91-C-0176; Mr. Michael Harris was the contracting officer's technical representative.

**Members of the AIAA.

ω Circular frequency (rad/sec)

ω_{e_i} A temporal circular encounter frequency in units of reciprocal time corresponding to the direction of $\bar{V}_w$ (rad/sec)

ω_o ϵ/k

LIST OF SUBSCRIPTS

i	Integer denoting orthogonal fluctuating velocity component (i = 1, 2, or 3)
k	Index representing GTOSL for the gust table whose immediate velocities are being calculated
l	Index representing GTOSL for the mutually correlated table of gust velocities that influence the gust table with index k
n	An integer

BACKGROUND

U.S. Navy rotorcraft have the unique problem of maneuvering for safe landing onboard ships that are rolling, pitching, and heaving and that are generating turbulent airwakes across the deck and downwind of the superstructure. To insure the compatibility of a particular rotorcraft and ship under various operating conditions (i.e., sea state, wind over deck, and weather), the Navy performs dynamic interface (DI) testing using fleet rotorcraft and various ships of opportunity. Since this approach is both costly and severely limited by the availability of fleet assets and the weather, the need has arisen to perform a greater portion of DI evaluations in the laboratory and to rely on actual testing for verification and training. Past efforts to simulate the airwake environment through scale-model wind-tunnel testing have correctly modeled the steady winds around the ship but produced questionable results with respect to ship-induced turbulence. Recent advances in the field of computational fluid dynamics (CFD) now make it possible to model the full-scale flow field around the ship, including turbulence, even for the interactive problem of a rotorcraft landing on a ship. Such CFD models can be used to show the effects of scale on turbulence and thereby correct currently deficient model-scale data. A computer code that can simulate this environment for the study of rotorcraft performance could be used to develop the best approach paths for safe landings, provide improved simulator training for pilots, and ultimately be used in the design or acceptance testing of future Navy rotorcraft. Such airwake simulation technology is also applicable to U.S. Army and civil rotorcraft nap-of-the-earth operations in the airwakes of terrain and buildings.

TECHNICAL OBJECTIVES

The technical objectives of this study were to outline approaches for modeling mathematically in real-time simulation the kinematics of a ship and the dynamics of a rotorcraft interacting aerodynamically within the ship's airwake on approach and landing aboard the moving ship. Specific computational methodologies for modeling the dynamics of the ship and rotorcraft and the associated aerodynamics are identified in this paper, and sufficient data is provided to demonstrate feasibility by example.

CANDIDATE TECHNICAL APPROACHES

There are at least three classes of identifiable options representing candidate technical approaches.

Option 1: Mutually Interactive Nonlinear Rotorcraft and Ship's Airwake Models, Including the Effects of Wind and Ship's Motion. This option not only models the aerodynamics of the atmospheric boundary layer interacting with the moving ship to form the ship's airwake and, in turn, models the aerodynamics of the ship's airwake interacting with the moving rotorcraft, but it also models the aerodynamics of the mutual interaction of the rotor downwash and outflow on the ship's moving deck and hangar boundaries, which modify the local airwake of the ship acting on the rotorcraft. This option, the most ambitious from a computational viewpoint, is especially applicable when the rotorcraft is transitioning to and from the lee of the ship's hangar in the presence of the ship's motions and varying wind-over-deck (WOD), because the rotor's downwash is impinging upon the ship's superstructure (e.g., deck, hangar door, etc.) and being recirculated back into the rotor. Mutually interactive computation in real-time simulation employing CFD modeling of rotorcraft and ship, however, appears not to be possible within the foreseeable future because of the extraordinary computational speed and time requirements. There is also the as-yet-unresolved issue of verifying the "correctness" of mutually interactive calculations, even in scaled time, with full-scale airwake measurements that do not intrinsically disturb the airwake. Since verifiable mutually interactive computation in scaled- or real-time simulation employing CFD modeling of rotorcraft and ship appears to be impractical, we believe that this is not a viable option at this time.

Option 2: Partially Interactive Nonlinear Rotorcraft and Ship's Airwake Models, Including the Effects of Wind and Ship's Motion. This option is valid in regions where the ship and the rotorcraft are only partially aerodynamically coupled by the influence of the ship's distributed airwake velocity field on the rotorcraft but NOT vice versa. Portions of this option have been developed and applied in real-time piloted simulations (Refs. 1 through 5). The extension of this option will be the main subject of this paper, because it is practical for real-time simulation within the foreseeable future, and because it is acceptable for approach and stationkeeping in the ship's airwake outside the lee of the hangar.

Option 3: Partially Interactive Linear Dynamic Rotorcraft Models and Steady-State Models of the Ship's Airwake and the Effects of Wind and Ship's Motion. This option is very useful for pre-experimental dynamic analysis of trimmed operating states, but it tends to be over simplified for real-time piloted simulation. The details of this option are presented in Ref. 6 and will not be discussed further in this paper.

REQUIRED MATHEMATICAL MODELS

The mathematical models that are required to simulate rotorcraft approach, stationkeeping, and landing on non-aviation ships include the kinematics, the dynamics, and the aerodynamics of the rotorcraft, the flight control system, the ship motion, and the airwake of the ship (which includes the mean winds, wind shears, and turbulence). Mathematical models of these components have been programmed and used for numerous years in real-time piloted simulations, and many are in the public domain. We propose to modify some of these models for the purpose of interfacing them with a newly developed model of the partially interactive rotorcraft and ship's airwake. The remainder of this paper contains brief descriptions of the extant models, the required modifications, and the new models. More detailed descriptions can be found in Ref. 6.

A. CFD MODEL OF SHIP AIRWAKE

A CFD model of the ship's airwake velocity field in the presence of specific ship's motions, WOD, and sea state is a prerequisite. Three-dimensional (3D) digital data bases for the turbulent kinetic energy (for which there are two equivalent symbols, TKE and k), the kinetic energy dissipation rate (ϵ), and each of the three orthogonal components of mean wind velocity are formed from the output of most CFD programs. For example, the CFD program called PHOENICS (Ref. 7) provides steady-state calculations of the airwake velocity fields around the ship (e.g., the ship is translating with respect to the airmass but not rotating). (Sample applications of PHOENICS for two different ships are provided in Refs. 8 and 9). Alternatively, the algebraic stress model of turbulence will provide variances and covariances of the three orthogonal components of fluctuating velocities. The results of the CFD modeling exercise are processed via a 3D sequential finite Fourier transform (FFT) program (see the subsequent discussion under Subtopic F) prior to the real-time simulation. Details of how the CFD program works can be found in Ref. 7. We will limit the discussion herein to how to use the outputs of the CFD program. We begin with a comparison of CFD and wind-tunnel data.

1. Comparison of Predicted and Measured Airwake Velocities

Comparison of the steady-state 1/80-scale airwake velocity field predictions for the DD963 "Spruance" class vessel by CFD in Ref. 9 with experimental 1/80-scale model airwake velocity measurements from a wind tunnel in Ref. 10 revealed good agreement throughout the airwake field except over and immediately aft of the landing deck. There was an experimental deficit of 86 percent (relative to the theoretical prediction by CFD) in mean longitudinal ($\bar{U}_{w/1}$) velocity component above the landing deck bulls-eye at hangar roof height with both a 20-kt and a 45-kt WOD from the bow, for example. There were also experimental deficits in ($\bar{U}_{w/1}$) of 42 and 48 percent at a point 30 cm (1/80 model scale) aft of the landing deck on the ship's centerline at hangar topside height with 20-kt and 45-kt WOD, respectively, from the bow. Excessive experimental multiples of TKE (again relative to predictions by CFD) occurred at the locations and WOD in Table 1. The cause of these excessive experimental values of TKE is attributed in Ref. 9 to predictable disturbing influences of the hot-wire anemometer probe and its mounting rake. These influences could be eliminated by employing laser Doppler anemometry in the wind tunnel.

The findings presented in Table 1 are supported by the qualitative results of the real-time piloted simulations reported in Ref. 1. The pilots judged the magnitude of the Ref. 10 turbulence to be excessive. When the variance of the turbulence* was reduced by factors of 2.5 to 3.3, the pilots judged the turbulence to be much more realistic.

2. Derivation of Energy Spectral Density from CFD Airwake Turbulence Model

The CFD airwake velocity predictions in Refs. 8 and 9 employed the "k, ϵ" turbulence model. The predicted values of turbulent kinetic energy k (L^2/T^2) and its dissipation rate ϵ (L^2/T^3) are used to calculate the root-mean-squared (RMS) values of velocity fluctuations and the frequency of the energy containing eddies. Assuming locally isotropic conditions, the RMS value of the turbulent wind-velocity fluctuations is given by:

$$\sqrt{\overline{u'^2}} = \left(\frac{2}{3}k\right)^{1/2} \qquad (1)$$

The characteristic circular frequency, ω_e (units of reciprocal time) of the energy-containing eddies is estimated from:

$$\omega_e \doteq \frac{\epsilon}{k} \qquad (2)$$

The power and spatial frequency of the locally isotropic turbulent energy spectral density model are scaled in accord with the predicted turbulent kinetic energy and its time rate of dissipation for the energy-containing eddies as described in general in Ref. 6 to characterize the energy spectral density of the turbulence.

There is some concern in the recommendations in Refs. 8 and 9 that "the time scale of turbulence (k/ϵ) in the CFD calculations may not be small compared to the time-scale of ship motion, and thus the (k, ϵ) model of turbulence may have to be modified," or the algebraic stress model of turbulence may have to be adopted.

B. ROTORCRAFT MODEL

An example of a generic single main rotorcraft model is the nonlinear Sikorsky-NASA GENHEL model (Ref. 11), which is used for real-time simulation but does not yet accept results of airwake CFD, even for batch simulation in scaled time. Rotorcraft distributed inflow, downwash, and sidewash aerodynamics interact with "deck-and-hangar" effects, but do not interact mutually with the ship's airwake. Rotor flapping, coning, lagging, air mass, and hub angular velocity degrees of freedom, as well as six degrees of fuselage rigid body and sling load freedom, are provided. The generic model is available in the public domain.

The rotorcraft is divided into components for the purpose of modeling the aerodynamics (e.g., the main rotor/hub, fuselage, lifting surfaces, and tail rotor). Each of these components comprises points that are located at some distance(s) relative to the center of gravity (c.g.) of the rotorcraft, and these components will be referred to as the distributed members of the rotorcraft.

The gust module in Ref. 11 applies the wind velocity components to the distributed members of the rotorcraft. The gust module does not, however, accept relative position-dependent components of the 3D mean-wind velocity field provided by CFD calculations of a ship's airwake. One purpose and subtopic of this paper is to describe a procedure for overcoming the foregoing limitation of accepting and applying mean-wind velocity components from CFD calculations.

A second purpose and subtopic of this paper is to recommend that the independent point correlated stochastic portion of the gust module in Ref. 11 be modified as recommended in Ref. 12 and further developed herein to provide both independent and mutually correlated "colored" randomly fluctuating velocities to the distributed members of the rotorcraft.

C. MEAN-WIND VELOCITY AND TURBULENT VELOCITY MODELS

The airwake of the ship is modeled by superimposing the three components of the mean velocity with the three components of the turbulence. The three components of the mean velocity, TKE and ϵ are computed at a point relative to the ship using the 3DFFT, which is discussed subsequently in Subtopic F. An overview of how the mean wind and turbulence models are interfaced with the rotorcraft aerodynamics model is shown in the computational flow diagram of Fig. 1. Some background on the mean wind and turbulence velocity models is presented in the following two subsections.

1. Mean Wind Velocity Model

To account for the distribution of mean-wind velocities over the rotorcraft, the mean-wind velocities will be calculated at all of the distributed members on the rotorcraft. Thus the location of all of the distributed members must be computed at each point in time relative to the ship. To appreciate the computational demands of this requirement, consider that the main rotor model of Ref. 11 is divided into N_{ss} segments for each of the N_B rotor blades, so that $2+(N_B)*(N_{ss})$ members will be required just for the main rotor, fuselage, and empennage. For each of the distributed members, three (3) orthogonal components of the mean-wind velocity associated with the location relative to the ship of that member will be computed using the truncated 3D sequential FFT technique described in Subtopic F. If $N_B = 4$ and $N_{ss} = 5$ the total number of parameter calculations for the main rotor, fuselage, and empennage models in each real-time loop interval is 3*(2+4*5)=66. This computational requirement is significant and may require a separate processor to allow for a reasonable duration of the real-time computation loop interval.

Since the 3D mean-wind velocity field provided by the CFD data base $(\underline{V}_{w/s})$ is usually defined with respect to a non-rotating ship, it must be transformed to the earth axes when the ship is rotating (roll, pitch, and yaw via the direction cosine matrix $T_{s/I}$) and translating with respect to the earth, at velocity $\underline{V}_{s/I}$, using the following vector equation:

$$\underline{V}_{w/I} = \underline{V}_{s/I} + T_{s/I}\underline{V}_{w/s} \qquad (3)$$

The mean winds computed in Eq. 3 are in the inertial axis system. In order to apply them to the appropriate rotor blade segment, they must be transformed into the blade axis system. These transformations must account for the orientation of the rotorcraft, the hub, and all of the rotor degrees of freedom. These complicated transformation are documented in Ref. 6 and will not be repeated here. The details of applying the mean winds at the other distributed members of the rotorcraft are also presented in Ref. 6.

2. Turbulent Velocity Model

a. Estimation of the Turbulent Energy Spectral Density in the Atmospheric Boundary Layer. Simulation of the turbulent velocity components is achieved by passing independent white noise processes having zero mean values and appropriate amplitude probability distributions through spectral filters whose transfer functions yield the desired forms of power spectral density. In order to make use of the spatially correlated values of TKE and ϵ from the CFD models, we must partition the locally isotropic energy PSD into the following components: (a) one-dimensional (1D) PSD functions of random turbulent velocity fluctuations at a single point in the flow field and (b) two-dimensional (2D) cross correlated PSD functions of random turbulent velocity fluctuations between two points in the flow field. An appropriate Dryden form for the point correlated one-dimensional PSD for the longitudinal velocity component of turbulence, derived in Appendix C of Ref. 12, is given in the first row of Table 2. The forms for the lateral and normal axes are presented in Ref. 6.

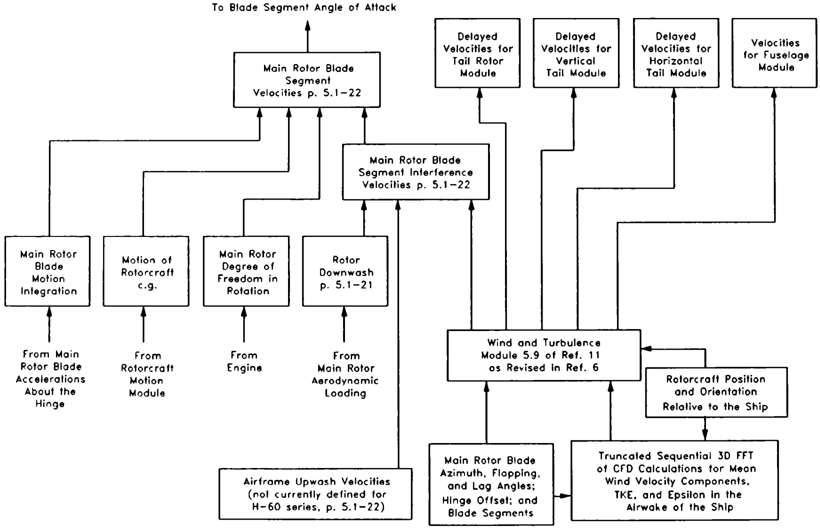
Appropriate forms for corresponding laterally correlated 2D PSD ratios, derived in Appendix D of Ref. 12, are given in the remaining rows of Table 2 and shown graphically in Fig. 2. The laterally correlated 2D PSD is a function of the (lateral) spatial separation distance, η, in units of length between points in the flow. Each PSD is expressed in terms of the spatial frequency (wave number) associated with the energy-containing eddies, Ω_e, the value

*The mean-squared value of locally isotropic turbulent velocity fluctuations is related to TKE by $\overline{u'^2} = (2/3)\ (TKE)$.

TABLE 1. EXCESSIVE EXPERIMENTAL MULTIPLES OF TURBULENT KINETIC ENERGY (TKE) FROM A COMPARISON OF REFS. 9 AND 10 WITH WOD FROM THE BOW OF A SCALE MODEL OF THE DD963 "SPRUANCE" CLASS VESSEL

Location	Height	Multiple in Experimental TKE Relative to CFD TKE	
		20 kt WOD	45 kt WOD
Port edge of landing deck athwart bulls-eye	Landing	5-fold	5 1/2-fold
Starboard edge of landing deck athwart bulls-eye	Landing	2-fold	2 1/2-fold
Port edge of landing deck athwart bulls-eye	Hangar Top Side	2-fold	3-fold
Starboard edge of landing deck athwart bulls-eye	Hangar Top Side	3-fold	2-fold
Port edge of landing deck athwart bulls-eye	Stack Up Takes	5 1/2-fold	5-fold
Starboard edge of landing deck athwart bulls-eye	Stack Up Takes	10-fold	11-fold
Port edge of after deck 30 cm* aft of bulls eye	Landing	4-fold	4-fold
Starboard edge of after deck 30 cm* aft of bulls eye	Landing	3-fold	4-fold
Port edge of after deck 30 cm* aft of bulls eye	Hangar Top Side	1 1/2-fold	4-fold
Starboard edge of after deck 30 cm* aft of bulls eye	Hangar Top Side	1 1/2-fold	3-fold

*1/80 model scale



To Blade Segment Angle of Attack
Main Rotor Blade Segment Velocities p. 5.1-22
Delayed Velocities for Tail Rotor Module
Delayed Velocities for Vertical Tail Module
Delayed Velocities for Horizontal Tail Module
Velocities for Fuselage Module
Main Rotor Blade Segment Interference Velocities p. 5.1-22
Main Rotor Blade Motion Integration
Motion of Rotorcraft c.g.
Main Rotor Degree of Freedom in Rotation
Rotor Downwash p. 5.1-21
From Main Rotor Blade Accelerations About the Hinge
From Rotorcraft Motion Module
From Engine
From Main Rotor Aerodynamic Loading
Wind and Turbulence Module 5.9 of Ref. 11 as Revised in Ref. 6
Rotorcraft Position and Orientation Relative to the Ship
Airframe Upwash Velocities (not currently defined for H-60 series, p. 5.1-22)
Main Rotor Blade Azimuth, Flapping, and Lag Angles; Hinge Offset; and Blade Segments
Truncated Sequential 3D FFT of CFD Calculations for Mean Wind Velocity Components, TKE, and Epsilon in the Airwake of the Ship

Figure 1. Computational Flow Diagram Showing Interfaces with the Wind and Turbulence Module Paragraph 5.9 of Ref. 11 as Revised in Ref. 6 (Page number citations are for Paragraph 5.1 of Ref. 11)

TABLE 2. ONE- AND TWO-DIMENSIONAL POWER SPECTRAL DENSITIES OF THE LONGITUDINAL AXIS COMPONENT OF TURBULENCE BASED ON THE DRYDEN FORMULATION*

	Longitudinal Fluctuating Velocity Component of Turbulence, u_1
One-Dimensional Power Spectral Density (PSD)**	$\overset{(1)}{\Phi}_{u_1u_1}(\Omega_1) = \dfrac{4B[E(\Omega_e)]\Omega_e^2}{32(\Omega_e^2 + \Omega_1^2)}$
Ratio of Laterally Correlated Two-Dimensional PSD to One-Dimensional PSD	$\dfrac{\overset{(2)}{\Phi}_{u_1u_1}(\Omega_1, \eta)}{\overset{(1)}{\Phi}_{u_1u_1}(\Omega_1)} = (\alpha\eta)^2 \left\{ \dfrac{K_1(\alpha\eta)}{\alpha\eta} - \dfrac{K_0(\alpha\eta)}{2} \right\}$ where $\alpha^2 = \Omega_e^2 + \Omega_1^2$
Low Frequency Asymptote of PSD Ratio	$\lim_{\alpha\eta \to 0} \dfrac{\overset{(2)}{\Phi}_{u_1u_1}(\Omega_1, \eta)}{\overset{(1)}{\Phi}_{u_1u_1}(\Omega_1)} = \Gamma(1) = 1$
High Frequency Asymptotes of PSD Ratio	$\to \sqrt{\dfrac{\pi}{2\alpha\eta}} e^{-\alpha\eta} \left[0.340 + 1.0625\alpha\eta - 0.5(\alpha\eta)^2 - \dfrac{\overset{0}{\underset{\vert}{0.0805}}}{\alpha\eta} + \ldots \right]$ If $\alpha\eta$ is large and $\vert arg\ \alpha\eta \vert < \frac{3}{2}\pi$ (this asymptote is plotted in Fig. 2)

*Higher order rational approximations of the one-dimensional von Karman spectra valid over a decade and a half above the half-power frequency are presented in Refs. 16 through 19.

**Mean-squared value of fluctuating velocity is defined as $\sigma_{u_1}^2 = 2\int_0^\infty \Phi_{u_1u_1}(\Omega_1)d\Omega_1$

$K_0(\alpha\eta)$ and $K_1(\alpha\eta)$ are modified Bessel functions of the second kind, order zero and one, respectively.

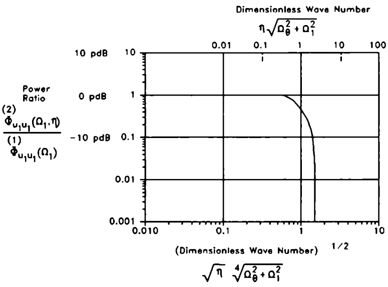


Figure 2. Laterally Correlated Two-Dimensional PSD Ratio of Longitudinal Gust Velocity

of the energy spectral density $E(\Omega)$ at $\Omega = \Omega_e$, i.e., $E(\Omega_e)$, and a dimensionless numerical scale factor, B. Values for Ω_e and $E(\Omega_e)$ are expressed in terms of turbulent kinetic energy, k, and its rate of dissipation, ϵ, in accord with the model of Ref. 13, as interpreted in Ref. 12. Values for B are subject to the constraint that

$$k = \int_0^\infty E(\Omega)d\Omega \qquad (4)$$

An example of the estimation of the wave number, Ω_e, associated with the energy-containing eddies in the wake of a backward-facing step is presented in Ref. 12. Reference 12 illustrates the conversion of dimensionless wave numbers $\eta\sqrt{\Omega_e^2 + \Omega_1^2}$ and $\eta\Omega_e$ to temporal circular encounter frequency ω_e, in the wake of a backward-facing step.

b. Rotating Ring Tables of Gust Velocity Components. To accommodate the longitudinal and lateral distribution of gust velocities over the helicopter, the addition of $2*(N_{ss})$ "rotating ring" tables is recommended to form a total of $1 + 2*(N_{ss})$ tables for each orthogonal velocity component that must be stored and updated in random access memory (RAM). The concept of a "rotating ring" table, introduced in Ref. 11, is described in Ref. 12. For $N_{ss} = 5$, as in Fig. 3, the resulting lateral array of eleven tables is depicted graphically in Fig. 4 as an overlay for Fig. 3. Starboard tables are assigned positive integers to represent GTOSL, the gust table offset laterally. Port tables are assigned negative integers to represent GTOSL. If GTOSL = 0, the gust table serves the hub aerodynamic calculations exactly as in the existing configuration of GENHEL (Ref. 11, ¶5.9).

Each of the three orthogonal components of random gust velocity will require $1 + 2*(N_{ss})$ tables corresponding to $1 + 2*(N_{ss})$ columns in the gust grid overlay, where N_{ss} = number of rotor blade segments/blade. The total number of tables, however, will be $3[1 + 2*(N_{ss})]$ if three random gust velocity components are applied. If N_{ss} = 5, thirty-three tables will be required in RAM. This computational requirement is also significant and may require one or more dedicated parallel processor(s).

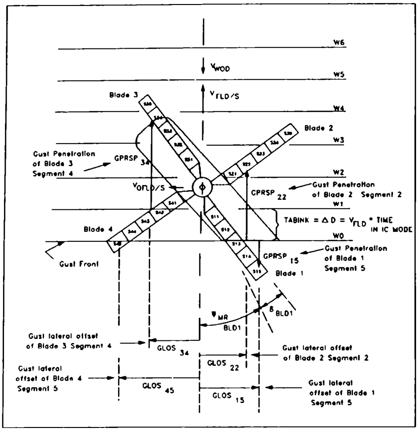

W6
V_{WOD}
W5
$V_{FLD/S}$
W4
Blade 3
Blade 2
W3
Gust Penetration of Blade 3 Segment 4
$GPRSP_{34}$
W2
$V_{OFLD/S}$
Gust Penetration of Blade 2 Segment 2
$GPRSP_{22}$
W1
TABINK = ΔD = V_{FLD} * TIME IN IC MODE
Blade 4
W0
Gust Front
Gust Penetration of Blade 1 Segment 5
$GPRSP_{15}$
Blade 1
$\Psi_{MR\,BLD1}$
δ_{BLD1}
Gust lateral offset of Blade 3 Segment 4
$GLOS_{34}$
Gust lateral offset of Blade 2 Segment 2
$GLOS_{22}$
Gust lateral offset of Blade 4 Segment 5
$GLOS_{45}$
$GLOS_{15}$
Gust lateral offset of Blade 1 Segment 5

Figure 3. Helicopter Penetration of Gust Front (adapted from Ref. 11)

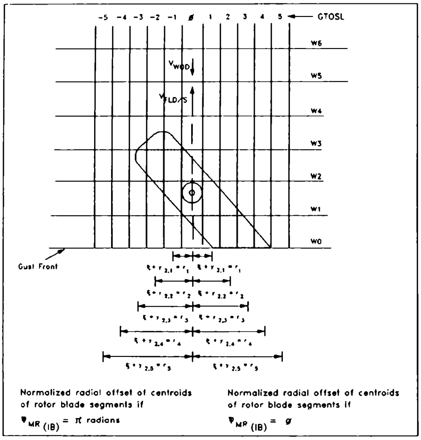

-5 -4 -3 -2 -1 ∅ 1 2 3 4 5 ← GTOSL
W6
V_{WOD}
W5
$V_{FLD/S}$
W4
W3
W2
W1
W0
Gust Front
$\xi + \gamma_{2,1} = r_1$ $\xi + \gamma_{2,1} = r_1$
$\xi + \gamma_{2,2} = r_2$ $\xi + \gamma_{2,2} = r_2$
$\xi + \gamma_{2,3} = r_3$ $\xi + \gamma_{2,3} = r_3$
$\xi + \gamma_{2,4} = r_4$ $\xi + \gamma_{2,4} = r_4$
$\xi + \gamma_{2,5} = r_5$ $\xi + \gamma_{2,5} = r_5$
Normalized radial offset of centroids of rotor blade segments if $\Psi_{MR\,(IB)} = \pi$ radians
Normalized radial offset of centroids of rotor blade segments if $\Psi_{MR\,(IB)} = \emptyset$

Figure 4. Two-Dimensional Laterally Correlated Gust Table Array Format [Overlay for Fig. 3]

The lateral array of 1D rotating ring gust tables depicted in Fig. 4 is always aligned with the wind direction relative to the ship and is oriented to surround the hub of the rotorcraft; i.e., the hub always resides in the column GTOSL = 0. To visualize the function of a 1D rotating ring table, consider the tread of a tank. As the tank translates longitudinally, progress is made by setting down segments of the tread before the tank and lifting them up behind, such that, although the tank is translating, each tread segment is fixed longitudinally. These tread segments are analogous to the cells in the 1D rotating ring gust tables, which are fixed longitudinally with respect to the ship.

The gust propagation velocity ($V_{FLD/S}$) is the component of the rotorcraft velocity relative to the ship in the direction of the WOD. In general, there exists a component of the rotorcraft velocity relative to the ship and orthogonal to the WOD. This component, denoted as $V_{OFLD/S}$ in Fig. 3, has the effect of translating the laterally correlated gust table arrays orthogonal to the direction of the WOD.

To accommodate the lateral translation of the gust tables, a separate 2D turbulence parameter table is used, denoted with dashed lines in Fig. 5. The columns of this turbulence parameter table, also aligned with wind over deck, are not laterally correlated and are not fixed laterally to the rotorcraft body as are the gust tables. Instead, the cells in this turbulence parameter table are fixed with respect to the ship. In effect, the turbulence parameter table is a 2D table.

The function of the 2D table may be visualized using a partially deflated basketball, the surface of which is divided into small segments. The contact patch of the basketball represents the active cells of the table. If the basketball is rolled in any direction, cells are added to the contact patch in the direction of travel and lifted from the other side, but the cells in the contact patch do not slip. Similarly, the cells in the turbulence parameter table are fixed with respect to the ship.

The turbulence parameter table generates the point correlated random gust velocities and, as directed by the table look-up pointer, transfers the gust velocities to the gust tables for lateral correlation and amplitude adjustment in accordance with relative gust power fraction distribution calculations to be discussed in the next subtopic.

The width of the columns in the turbulence parameter tables, X_{TPT}, is dependent on the magnitude of $V_{OFLD/S}$ and the width of the columns in the laterally correlated gust tables, X_{CT} by

$$X_{TPT} = |V_{OFLD/S}| * TIME + X_{CT} \qquad (5)$$

where

$$V_{OFLD/S} = V_{KT}(1.689)\sin\psi_{RWOD} \text{ (ft/sec)}$$

$$X_{CT} = R_{MR}\frac{(1-\xi)}{N_{SS}}$$

so that the minimum width of the columns in the turbulence parameter tables is equal to the width of the columns in the gust tables.

c. Assembling the Array of Random Noise Filters for Turbulence Modeling in Real Time. For each partitioned PSD function, it is necessary to convert dimensionless wave numbers to dimensional encounter frequencies and select Gaussian random noise filters for simulating the required PSDs of fluctuating velocity components at each spanwise station. Real-time difference equations are written for each filter, and the arrays of point correlated and mutually cross correlated filtered randomly fluctuating velocity components are summed. One such sum for GTOSL = -3 is depicted by a flow chart in Table 3. There will be $1 + 2*(N_{SS})$ such sums for each orthogonal fluctuating velocity component.velocity components are summed.

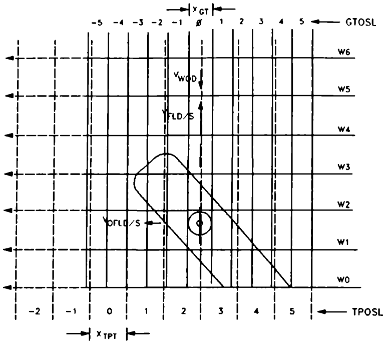


Figure 5. Two-Dimensional Laterally Correlated Gust Table Array Fixed Laterally to Rotor Hub from Fig. 4 with Solid Lines Shown Overlaying Laterally Offset Turbulence Parameter Table Fixed Laterally to Ship with Dashed Lines

* * * * * * * * * * * * * * *

To each of the $1 + 2^*(N_{ss})$ existing point correlated filtered "rotating ring" tables, there must be added the appropriate $2^*(N_{ss})$ additional velocities to represent mutually laterally correlated turbulent velocity contributions among the gust velocity tables represented by each value of GTOSL. The relative power fraction distribution $p_{u_{lkl}}$ among the $2^*(N_{ss})$ contributions to each of the $1 + 2^*(N_{ss})$ laterally offset gust tables can be determined for each orthogonal gust velocity component, e.g., u_1, as

This ratio is from Table 2, second row ↓

This PSD is from Table 2, first row ↓

$$p_{u_{lkl}} = \frac{\int_0^\infty \left[\frac{\overset{(2)}{\Phi}_{u_l u_l}(\Omega_l, |\eta_l - \eta_k|)}{\Phi_{u_l u_l}(\Omega_l)} \right] \Phi_{u_l u_l}(\Omega_l) d\Omega}{\int_0^\infty \Phi_{u_l u_l}(\Omega_l) d\Omega_l} \qquad (6)$$

↑ This integral is $\frac{\sigma^2_{u_l}}{2}$, if based on Table 2

where l is a subscript representing the orthogonal fluctuating velocity component, k is an index representing GTOSL for the gust table whose immediate point correlated and laterally correlated velocities are being calculated and added, and l is an index representing GTOSL for the mutually correlated table of gust velocities that influence the gust table with index k.

The relative power fraction $p_{u_{lkl}}$ in Eq. 6 is plotted in Fig. 6 as a function of the dimensionless lateral separation parameter $\Omega_0|\eta_l - \eta_k|$ defined in Fig. 6. For the example of a 2D backward-facing step in Appendix B in Ref. 6, Ω_0 = 1.5/h units of reciprocal length, and the dimensionless abscissa in Fig. 6 can be interpreted as $(|\eta_l - \eta_k|)/h$, where h is the height of the step.

For each index, k, the describing function $H_{u_{lk}}$ of the sum in Table 4 of the independent random and mutually correlated fluctuating velocities will be

$$H_{u_{lk}}(j\omega) = \sqrt{f_{pu_l}}[F_{u_{lk}}(j\omega)]\left\{1 + \sum_l (1 - \delta_{kl})[C_{u_{lkl}}(j\omega, |\eta_l - \eta_k|)]\right\} \qquad (7)$$

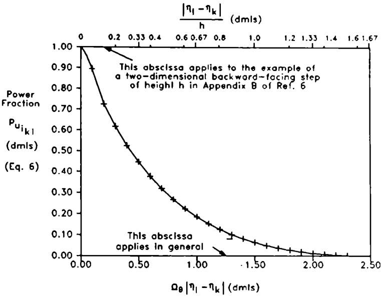
where

δ_{kl} is the Kronecker delta = $\begin{cases} 1; l = k \\ 0; l \neq k \end{cases}$

f_{pu_l} is the overall power fraction whose reciprocal is given by

$$(f_{pu_l})^{-1} = \sum_k \left\{ 1 + \sum_l (1 - \delta_{kl}) p_{u_{lkl}} \right\} \qquad (8)$$

for each of the original $1 + 2^*(N_{ss})$ independent random number generators and point correlated filters, $F_{u_{lk}}(j\omega)$ for velocity component u_l



Ω_0 = characteristic spatial wave number (reciprocal length) associated with the energy-containing eddies in a turbulent boundary layer representing an airwake of a vessel, structure, or topographic feature

$|\eta_l - \eta_k|$ = magnitude of lateral separation (length) between two transverse points in the (longitudinal) turbulent wake

h = height of two-dimensional backward-facing step

Figure 6. Laterally Correlated Power Fraction for Longitudinal Fluctuating Velocity Component of Turbulence (u_1) based on the Dryden Formulation of Power Spectral Density

* * * * * * * * * *

The example of a 2D backward-facing step of height h in Appendix B of Ref. 6 can be used to illustrate the calculation of f_{pu_l} in Table 4 for N_{ss} = 5 and R_{WR} = h = 25 ft using Eq. 8. $(|\eta_l - \eta_k|)/h$ can, in turn, take on the $1 + 2^*(N_{ss})$ = 11 values: 0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0 (dmls) among the GTOSL for which the corresponding power fractions $(p_{u_{lkl}})$ are from Fig. 6: 1.0, 0.61, 0.38, 0.22, 0.13, 0.06, 0.02, 0.01, 0.0, 0.0, 0.0 (dmls). The resulting overall power fraction f_{pu_l} equals 0.0275 for this example. It is necessary to apply the amplitude fraction $\sqrt{f_{pu_l}}$ = 0.1658 to all filters $F_{u_{lk}}(j\omega)$ for one-dimensional power spectral densities in each GTOSL.

This concludes our discussion of a model for the turbulent velocity distribution acting on the main rotor in the airwake of the ship.

D. FLIGHT CONTROL SYSTEM

Mathematical models of the flight control system used for real-time piloted simulation of the SH-60B Sea Hawk helicopter during shipboard approach and landing are presented in Ref. 1. The features of the flight control system include: (a) rate damping in the roll, pitch, yaw, and heave axes [usually called the stability augmentation system (SAS)]; (b) attitude-command, attitude-hold capabilities in the pitch and roll axes [usually activated by the "autopilot" switch on the automatic flight control

TABLE 3. EXAMPLES OF RANDOM NOISE FILTERING PATHS AMONG GUST TABLE OFFSETS LATERALLY (GTOSL) TO PRODUCE ONE SUM OF LONGITUDINAL GUST VELOCITY SIGNALS (u_i) AT GTOSL=-3

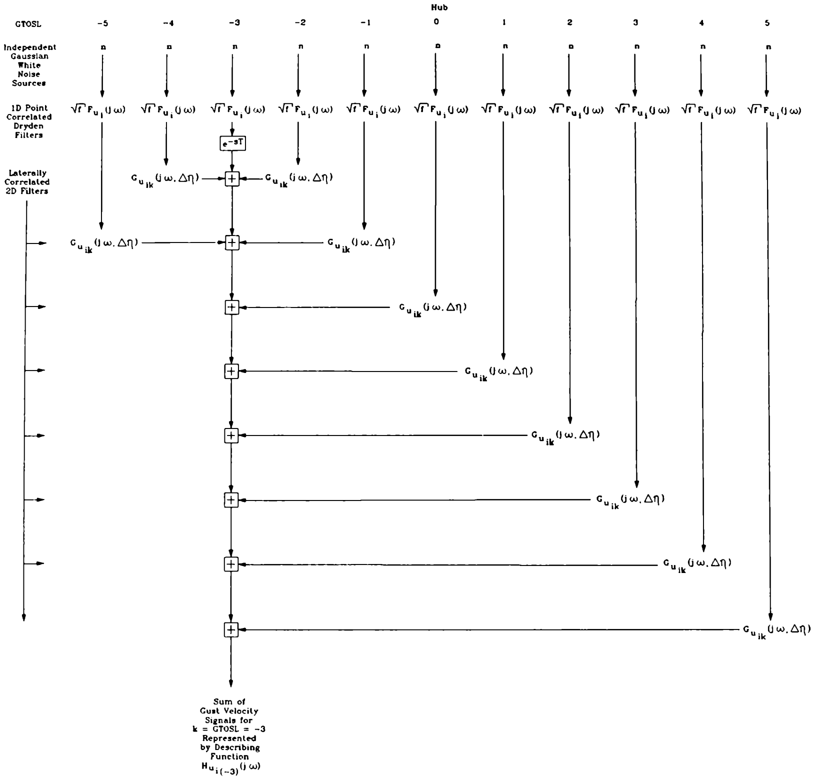

Hub
GTOSL
-5
-4
-3
-2
-1
0
1
2
3
4
5
Independent Gaussian White Noise Sources
n
1D Point Correlated Dryden Filters
$\sqrt{\Gamma} F_{u_i}(j\omega)$
e^{-sT}
Laterally Correlated 2D Filters
$G_{u_{ik}}(j\omega, \Delta\eta)$
Sum of Gust Velocity Signals for k = GTOSL = -3 Represented by Describing Function $H_{u_{i(-3)}}(j\omega)$

TABLE 4. POWER FRACTION DISTRIBUTION OF $P_{u_{lk}}$, AMONG "ROTATING RING" GTOSL FOR A SINGLE MAIN ROTOR IN THE AIRWAKE OF A TWO-DIMENSIONAL BACKWARD FACING STEP OF HEIGHT $h = R_{ur}$ WITH $N_{ss} = 5$ FOR USE WITH EQ. 8

GTOSL Index k → Index l↓	-5	-4	-3	-2	-1	0	1	2	3	4	5
-5	1	0.61	0.38	0.22	0.13	0.06	0.02	0.01	0	0	0
-4	0.61	1	0.61	0.38	0.22	0.13	0.06	0.02	0.01	0	0
-3	0.38	0.61	1	0.61	0.38	0.22	0.13	0.06	0.02	0.01	0
-2	0.22	0.38	0.61	1	0.61	0.38	0.22	0.13	0.06	0.02	0.01
-1	0.13	0.22	0.38	0.61	1	0.61	0.38	0.22	0.13	0.06	0.02
0	0.06	0.13	0.22	0.38	0.61	1	0.61	0.38	0.22	0.13	0.06
1	0.02	0.06	0.13	0.22	0.38	0.61	1	0.61	0.38	0.22	0.13
2	0.01	0.02	0.06	0.13	0.22	0.38	0.61	1	0.61	0.38	0.22
3	0	0.01	0.02	0.06	0.13	0.22	0.38	0.61	1	0.61	0.38
4	0	0	0.01	0.02	0.06	0.13	0.22	0.38	0.61	1	0.61
5	0	0	0	0.01	0.02	0.06	0.13	0.22	0.38	0.61	1
Column Sums	2.43	3.04	3.42	3.64	3.76	3.80	3.76	3.64	3.42	3.04	2.43

NOTES: The diagonal elements with value "1" represent the point correlated one-dimensional power fraction

Columns above and below the diagonal elements with value "1" contain values representing laterally correlated two-dimensional power fractions

To find the reduced overall value, f_{pu_l}, for the point correlated one-dimensional power fraction by Eq. 8, compute the sum of column sums and reciprocate, because the sum of sums must be unity by definition over the surface of the main rotor

$$f_{pu_l} = \frac{1}{36.38} = 0.0275 \text{ (dmls)}$$

Apply the amplitude fraction $\sqrt{f_{pu_l}}$ = 0.1658 (dmls) to all filters $F_{u_{lk}}(j\omega)$ for one-dimensional power spectral densities in each GTOSL.

system (AFCS) control panel]; (c) rate-command, attitude-retention capabilities in the yaw axis; and (d) trim release capability in the roll, pitch, yaw, and collective axes*. A detailed description of the SH-60B flight control system can be found in Ref. 14.

E. SHIP MOTION MODEL

The ship's roll, pitch, yaw, surge, sway, and heave motions are functions of the ship's heading and sea state. For example, thirty-sinusoid models of each of the six degrees of freedom are applied to batch simulation in scaled time in Ref. 15, and six-sinusoid models of each of six degrees of freedom are applied to DI simulation in real time in Ref. 1.

F. THREE-DIMENSIONAL SHIP CORRELATED AIRWAKE VIA 3DFFT

1. Non-Rotating Ship

In order to use the 3D CFD information in a real-time simulation, a continuous representation of these digital data bases is required. This is accomplished by approximating a digital 3D data base with a continuous function of spatial position using a truncated 3D sequential FFT technique, as described below.

The truncated 3D sequential FFT technique consists of computing sequentially the FFT of arrays of data from the CFD digital data bases. First, the FFT of all of the vectors of points in the data bases with like (j,k) indices** is computed, resulting in a continuous approximation of each vector. Second, the FFT of the vectors of coefficients resulting from the first FFT computation for a given (k) index is computed. This step results in a continuous approximation of each surface in the data base for a given (k) index. Next, the FFT of each of the vectors of coefficients created in the second step of FFT computation is computed, resulting in a continuous 3D approximation of the data base. This procedure is called wind flow field data base decomposition and is accomplished with the aid of the proprietary FREDA (FREquency Domain Analysis) computer program in advance of real-time simulation. Finally, the matrices of complex Fourier coefficients resulting from the data base decomposition are stored for subsequent use by the mean wind velocity-and-turbulence parameter calculation program during the real-time simulation.

To demonstrate briefly how this technique works, Eq. 9 gives the expression for the axial component of wind over deck as a function of spatial position:

$$V_{x_{w/s}}(x,y,z) = \Big\{ C_0(y,z) + C_1(y,z)^* x + \sum_{i=1}^{MX} [A_i(y,z)\sin\omega_i x + B_i(y,z)\cos\omega_i x] \Big\}_{V_{x_{w/s}}} \tag{9}$$

where $\omega_i = (2\pi i)/(NX)_{V_{x_{w/s}}}$ for $(NX)_{V_{x_{w/s}}}$ points in the $V_{x_{w/s}}$ data base, and $MX \le NX/2$ for the indicated truncated series.

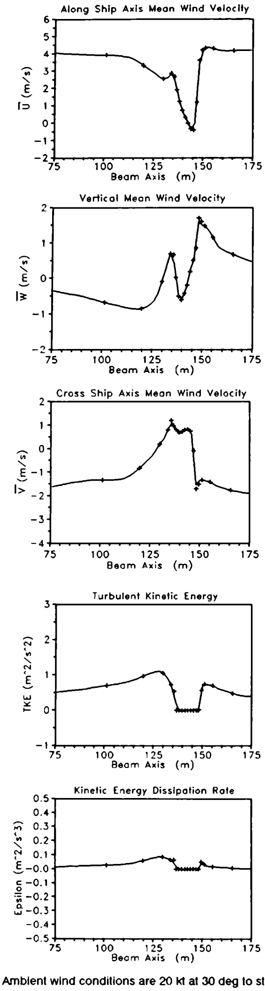
Similar expressions exist for $V_{y_{w/s}}(x,y,z)$, $V_{z_{w/s}}(x,y,z)$, $TKE_{w/s}(x,y,z)$, and $\epsilon_{w/s}(x,y,z)$ and are presented in Ref. 6. Scalar functions C_0, C_1, A_i, and B_i for each of the three components are defined in Appendix B of Ref. 6. A thorough description of this mathematical modeling technique applied to the DD963 CFD data base in Ref. 9 is also provided in Ref. 6. Some sample results for the DD963 CFD data base are presented in Fig. 7.

2. Rotating Ship

The effect of ship rotation on the airwake velocity field involves transient CFD calculations with time-varying geometry of the ship's boundaries; whereas, the past calculations for the DD963 in Ref. 9 and the LHA in Ref. 8 have been for non-rotating ships. "Since a transient CFD calculation proceeds as a sequence of 'steady-state' iterative calculations, one at each time-step, the computer time requirements increase substantially" (Refs. 8 and 9) for a moving ship.

*A collective proportional force gradient with a trim release switch is standard equipment on the Navy's SH-60B Sea Hawk.

**Array index k is not to be confused with turbulent kinetic energy k.



Notes:
1. Ambient wind conditions are 20 kt at 30 deg to starboard
2. Ship's port and starboard gunwales are at beam locations of 135 and 150 m, respectively
3. The "test" location is 5 m aft of the hangar door and 5 m above the landing pad

Figure 7. Sample Results of Applying the 3DFFT to the DD963 CFD Data Base

A means of incorporating the effect of ship motion on the airwake velocity field has been devised. A number of CFD data bases modeling the steady-state airwake velocity field about a ship inclined at varying pitch and roll angles could be generated. The pitch angle of the ship could then be thought of as a fourth dimension for the airwake flow field, and the roll angle, as the fifth. In other words, it would take five quantities to define the flow field parameters: given the longitudinal, lateral, and vertical location within the flow field and the ship's pitch and roll angles, a unique set of airwake flow field parameters may be determined. These extra dimensions would append to the truncated 3D sequential FFT technique, thereby creating a truncated five-dimensional (5D) sequential FFT technique. The 3D technique described above would be performed on all of the CFD data bases for the varying pitch and roll angles of the ship. Then, the FFT of all of the vectors of coefficients resulting from the third step of the 3D process associated with data bases with like roll angles would be computed. Finally, the FFT of the vectors of coefficients generated by this fourth step of FFT computation would be computed. This would create a continuous 5D function for each flow parameter.

The appropriate ship pitch and roll angles used to determine the values of the flow field parameters would depend on the position of the rotorcraft in the airwake velocity field. In the lee of the pitching and rolling ship, an estimate of the airwake velocity field is a mixture of the steady-state velocity fields for a number of static ship orientations. For example, if the mean air velocity relative to the ship is 20 m/sec, an estimate of the airwake velocity field at a distance of 40 m from the ship in the leeward direction would be the steady-state velocity field about a ship whose pitch and roll angles are equal to the actual ship's pitch and roll angles at a time 2 sec in the past. This is due to the fact that the air affecting the flow field at this location passed over the ship 2 sec prior to reaching the location of interest. In general, therefore, the appropriate ship's pitch and roll angles used to calculate the flow field parameters at a given location in the airwake velocity field correspond to the orientation of the ship at the time the air at that location in the flow field passed over the ship. This time would be determined by first establishing the distance from the given location to the ship in the leeward direction and then dividing that distance by the mean-wind velocity.

CONCLUSIONS

The technical objectives of this study were to outline approaches for modeling mathematically in real-time simulation the kinematics of a ship and the dynamics of a rotorcraft interacting aerodynamically within the ship's airwake on approach and landing aboard the moving ship. These objectives have been achieved. The resulting ship-correlated airwake, wind, and turbulence velocity models are intended for use in real-time simulation with a complete rotorcraft mathematical model, as detailed herein, incorporating distributed rotor blade elements. Non-uniform distributions of deterministic wind velocities and turbulence correlated with specific features of the ship are represented in the airwake disturbance model.

REFERENCES

1. Clement, Warren F., The Study of Helicopter Flying Qualities Requirements for Shipboard Operations. Vol. I: Analytical Study; Clement, Warren F., and Wayne F. Jewell, Vol. II: Piloted Simulation Study; Jewell, Wayne F., Bimal L. Aponso, and Warren F. Clement, Vol. III: Mathematical Models Used for the Piloted Simulation, NADC-82150-60, Three Volumes, July 1986.

2. Stapleford, Robert L., Warren F. Clement, Robert K. Heffley, George C. Booth, and Robert L. Fortenbaugh, Flight Control/Flying Qualities Investigation for Lift/Cruise Fan V/STOL. Vol. I: Analytical Development; Vol. II: Piloted Simulation; Vol. III: Simulator Model, NADC-77143-30, August 1979.

3. Ringland, Robert F., and Warren F. Clement, Systems Definition Phase NAVTOLAND Rotary Wing Program, SH-2F Control Laws Development. Vol. I: Analytical Development; Vol. II: Simulation Data, Systems Technology, Inc., TR-1177-1, April 1982.

4. Ringland, Robert F., and Susan A. Riedel, Type A V/STOL Simulation Equations and Data, Systems Technology, Inc., TR-1128-1, July 1979.

5. Ringland, Robert F., and Wayne F. Jewell, Type A V/STOL Simulation Equations and Data, Systems Technology, Inc., TR-1148-1, December 1980.

6. Clement, Warren F., and Peter J. Gorder, Modeling of Rotorcraft and Ship Dynamic Interface, Systems Technology, Inc., TR-1288-1, March 1992.

7. Spalding, D. B., PHOENICS 1984--A Multidimensional Multi-Phase General-Purpose Computer Simulator for Fluid Flow, Heat Transfer, and Combustion, CHAM Report CFD/84/18, 1984.

8. Mahaffey, W. A., Turbulent Ship Airwake Environment Prediction Methodology, Final Report, CHAM of North America, Inc., Report 4031/23, September 1987.

9. Mahaffey, W. A., and C. E. Smith, Turbulent Ship Airwake Environment Prediction Methodology, Interim Report, CHAM of North America, Inc., Report 4031/19, March 1987.

10. Garnett, T. S., Jr., Investigation to Study the Aerodynamic Ship Wake Turbulence Generated by a DD-963 Destroyer, NADC Report No. 77-214-30, March 1980.

11. Howlett, J. J., UH-60A Blackhawk Engineering Simulation Program; Vol. I: Mathematical Model; Vol. II: Background Report, NASA CR-166309, December 1981, Rev. 1 by Mark Ballin, August 1988.

12. Clement, Warren F., Wayne F. Jewell, Peter J. Gorder, Urban Svensson, Kyna Schutzback, and Nicholas S. Vlachos, Atmospheric Turbulence Modeling for Real-Time Simulation of Nap-of-the-Earth (NOE) Flight, Systems Technology, Inc., Technical Report No. 1266-1, Aeroflightdynamics Laboratory, U.S. Army Aviation Research and Technology Activity, U.S. Army Aviation Systems Command, Report TR-89-A-004, May 1989.

13. Rhodes, N., and D. Kirkcaldy, Estimating Spectral Distribution of Turbulence, CHAM Report No. 2793/1, 1984.

14. Cable, LCDR L. G., and Mr. J. H. Edris, SH-60B Helicopter Criteria for the Device 2F135 Operational Flight Trainer, Naval Air Test Center Technical Report RW-2R-82, July 1, 1982.

15. Clement, Warren F., and Wayne F. Jewell, Predicting Shipboard Stationkeeping and Landing Precision of Rotorcraft. Vol. I: Executive Summary, Vol. II: Technical Approach, Vol. III: Appendices, Systems Technology, Inc., TR-2306-1, June 1987.

16. Campbell, C. Warren, A Spatial Model of Wind Shear and Turbulence for Flight Simulation. Ph.D. Thesis--Colorado State University, NASA TP-2313, May 1984.

17. Campbell, C. Warren, Adding Computationally Efficient Realism to Monte Carlo Turbulence Simulation, NASA TP-2469, May 1985.

18. Campbell, C. W., and G. H. Fichtl, "Recent Advances in Monte Carlo Turbulence Simulation," Proc. Conf. on Aerospace and Range Meteorology, Huntsville, AL, August 27-29, 1985, Amer. Met. Soc., pp. 65-68.

19. Campbell, C. Warren, "Monte Carlo Turbulence Simulation Using Rational Approximations to von Karman Spectra," Approximate Simulation of Turbulence, NASA Tech Briefs MFS-28172, circa 1985.

AIAA-92-4150-CP

ANALYSIS OF PROPULSION SYSTEM DYNAMICS IN THE VALIDATION OF A HIGH-ORDER STATE SPACE MODEL OF THE UH-60

Frederick D. Kim
NASA Ames Research Center
Moffett Field, California

Abstract

Frequency responses generated from a high-order linear model of the UH-60 Black Hawk have shown that the propulsion system influences significantly the vertical and yaw dynamics of the aircraft at frequencies important to high-bandwidth control law designs. The inclusion of the propulsion system comprises the latest step in the development of a high-order linear model of the UH-60 that models additionally the dynamics of the fuselage, rotor, and inflow. A complete validation study of the linear model is presented in the frequency domain for both on-axis and off-axis coupled responses in the hover flight condition, and on-axis responses for forward speeds of 80 and 120 knots.

Nomenclature

A, **B**, **C**	Linearized system matrices
a_{ij}, b_{ij}	Elements of **A** and **B**
B_3	Compressor-diffusor bleed fraction
J_z	Moment of inertia of the rotating components aligned along the vertical axis, slugs-ft^2
J_{GT}	Moment of inertia of the gas generator, compressor, and shafting, slugs-ft^2
J_{tot}	Rotational inertia of all components powered by the engine, slugs-ft^2
K_C	Collective anticipation gain, lbm/in-sec
K_D	Fuel control derivative gain, lbm-sec
K_I	Fuel control integral gain lbm/sec
K_P	Fuel control proportional gain, lbm
K_{bl}	Fraction of diffusor bleed gas used to cool the gas generator turbine blades
K_{V_3}	Compressor volume coefficient
$K_{V_{41}}$	Gas generator volume coefficient
$K_{V_{45}}$	Power turbine volume coefficient
N_G	Gas generator speed, RPM
P_3	Compressor discharge pressure, psi
P_{41}	Gas generator inlet pressure, psi
P_{45}	Power turbine inlet pressure, psi
p, q, r	Angular rate components, rad/sec
Q_C	Torque required by compressor, ft-lb
Q_e	Torque produced by the engines, ft-lb
Q_{GT}	Torque output of the gas generator, ft-lb
Q_{PT}	Torque output of the power turbine, ft-lb
Q_{req}	Total required torque, ft-lb
T_F	Fuel control time constant, sec
T_3	Compressor discharge temperature, °R
T_{41}	Gas generator turbine inlet temperature, °R
T_{45}	Power turbine inlet temperature, °R
u, **x**, **y**	Control, state, and output vectors
u, v, w	Velocity components, ft/sec
W_{A_3}	Compressor discharge mass flow rate, lbm/sec
$W_{A_{31}}$	Combustor intake mass flow rate, lbm/sec
W_{B_3}	Diffusor bleed discharge flow rate, lbm/sec
W_1	Fuel flow change due to RPM error, lbm/sec
W_2	Fuel flow change due to collective, lbm/sec
W_{41}	Gas turbine intake mass flow rate, lbm/sec
W_{45}	Power turbine intake mass flow rate, lbm/sec
W_f	Fuel flow rate, lbm/sec
β_0	Collective flap angle, rad
β_{1c}, β_{1s}	Cyclic flap angle, rad
β_2	Reactionless flap angle, rad
Δt	Integration step size, sec
δ_{col}	Collective stick input, in
δ_{lat}	Lateral stick input, in
δ_{lon}	Longitudinal stick input, in
δ_{ped}	Pedal input, in
δ_{stab}	Horizontal stabilator position, deg
ζ_0	Collective lag angle, rad
ζ_{1c}, ζ_{1s}	Cyclic lag angle, rad
ζ_2	Reactionless lag angle, rad
ϕ_{DYN}	Elastic twist at main rotor blade tip, rad
ν_0	Uniform main rotor induced velocity
ν_{1c}	Cosine of main rotor induced velocity
ν_{1s}	Sine of main rotor induced velocity
ν_t	Tail rotor induced velocity
ν_x, ν_y	Delayed fuselage wake
Φ, Θ, Ψ	Aircraft Euler angles, rad
ψ	Blade azimuth angle, rad
Ω	Main rotor speed, rad/sec

Introduction

In response to the rising demands for highly agile and maneuverable aircraft, the handling qualities specifications have become more stringent in recent years [1], and it is becoming increasingly obvious that in order to satisfy these demands, these aircraft will require high-bandwidth, high-authority flight control systems to reduce off-axis couplings, increase agility, and provide the desired response modes for given missions.

The high bandwidth designs extend the dynamics of the flight control systems to couple with the high frequency dynamics of the rotor, inflow, actuator, and the propulsion system. This coupling of airframe and control system dynamics limits the maximum usable gains of a flight control system and when excluded in the design can trigger closed loop instabilities [2, 3]. Refs. [4, 5] present a mathematical model of a helicopter that is capable of generating high-order linear models of the UH-60 in all steady state flight conditions including coordinated turns, climbs, and descents. The model accounts for dynamics of the rotor, inflow, and the actuators and has been used successfully in quantifying the effects of rotor dynamics on control law designs [6]. However, the model assumes constant rotor speed and does not include the effects of the propulsion system dynamics.

Propulsion system dynamics needs to be considered, because the dynamics of the fuel control system, gas generator, and power turbine is well within the frequency range of modern flight control systems. Furthermore, both the fuel control system and the power turbine dynamics are coupled to the main rotor dynamics through the rotor speed degree of freedom, and their interaction must be analyzed together to give an accurate prediction of their dynamic response.

Coupled rotor and propulsion system analysis originated with the need to evaluate the first torsional mode of the rotor/driveshaft, which can be described as an oscillation where the main rotor blades lag together against the motion of the main rotor drive shaft. Refs. [7, 8] investigate this motion using a three degree of freedom engine-rotor model represented by rotating inertias connected with springs and dampers. This analysis was used successfully to reproduce and subsequently to attenuate the torque and fuel flow oscillations of the CH-47C [8].

In Ref. [9], Kuczynski *et al.* coupled a comprehensive nonlinear engine and fuel control model to a detailed airframe and rotor simulation program developed at Sikorsky [10] and formulated a model that was valid up to frequencies of 6 Hz. The nonlinear model was used to investigate how the choice of engine/fuel control parameters influence transient rotor speed droop as well as directional handling qualities, particularly yaw damping, directional control power, and Dutch roll. The model contains all the high-order dynamics necessary for accurate control designs, but a linearized model is not presented in the paper.

In Ref. [11], Hull presents a coupled analysis of the rotor and the propulsion system for a generic helicopter. The dynamics of the rotor, fuselage, engine, fuel control, drive train, lag damper, and the tail rotor are included in his mathematical model. The linear model includes a four-state perturbation model of the engine, a drive train consisting of torsional springs, dampers, and inertias, and a fuselage-rotor model extracted from a nonlinear time history simulation using a stepwise regression technique. The complete linear model is of sufficiently high-order to study the effects of propulsion system dynamics on advanced control law designs, but when validated against a high-fidelity UH-60 simulation of Ref. [10], the agreement was very poor.

Ockier, in Ref. [12], studied the dynamic interaction of the engine drive train, and a four-bladed rotor model. His engine model contains four states and is a first-order state space version of component type developed in Ref. [13]. A second order fuel control model consisting of a proportional-plus-integral-plus-derivative (PID) controller is used. The drive train is assumed to be rigid and hence the flexibility of the rotor shaft is neglected, and the power turbine speed and the rotor speed are related simply by a gear ratio. The rotor model is formulated with flap, lag, and inflow degrees of freedom and is valid only for hover. The complete linear model consists of 25 states and is used to demonstrate in detail the coupled rotor speed and collective lag oscillations.

In the literature, the highest order linear model that includes the effects of the propulsion system is that of Kaplita *et al.* [14]. Their linear model of the CH-53 helicopter has a total of 42 states and is applicable to both hover and forward flight conditions. It retains the gas generator speed degree of freedom from the propulsion system. The power turbine is related by a gear ratio to the main rotor speed, which has its own degree of freedom. Fuel control dynamics is neglected. Instead, the fuel control is modeled as an input to the system. The rotor model includes flap, lag, and inflow degrees of freedom and is based on the generic model given in Ref. [10]. Additional dynamics considered in the model are the dynamics of suspended load and three elastic bending modes of the fuselage.

None of the mathematical models reviewed has all the ingredients for use as a complete linear model for a credible linear control law design and development. Ref. [14] appears to be the most applicable, but a major limitation of this linear model is that because the dynamics of fuel control system is not considered, the frequency response of the helicopter does not contain the closed-loop response of the propulsion system. As a result, an effort to extend the model of Refs. [4, 5] was conducted at NASA to extract a high-order linear model that includes the dynamics of the propulsion system as well as that of the fuselage, rotor, inflow, and the actuators. This model has several advantages over that of Ref. [14] in that it is built around a more theoretically rigorous coupled trim algorithm, in which force and moment equilibrium, as well as several other kinematic and rotor dynamic response conditions are explicitly enforced, and in which the full periodicity of both rotor and fuselage response is rigorously maintained. Furthermore, in the model of Refs. [4, 5], both the solution and linearization of the equations of motion are carried out simultaneously for all of its dynamic components.

The primary goal is to combine this model with the first-order state space formulation of the T700 that was developed in Ref. [12] to give a coupled fuselage, rotor, and propulsion system analysis. The combined model retains the PID controller model structure of Ref. [12] with the gains and time constants identified from flight test frequency responses. This approach was chosen because a purely analytical model of the electrical control unit of the T700 engine is overly complex. It contains numerous time delays, feedback loops, and hysteresis effects and is not easily applicable to existing high-order linear models of the UH-60. On the other hand, a PID model of the controller is very simple and provided the identification is conducted in the valid frequency range, it can model the coupling dynamics of the rotor and the propulsion system.

Therefore, the objectives of this paper are (i) to present frequency domain techniques to identify the dynamics of a simplified fuel control system, (ii) to assess the significance of including the high-order effects of the propulsion system dynamics on the predicted aircraft dynamics and (iii) to present frequency response validation results of a high-order linear model of the UH-60 that includes the dynamics of fuselage, rotor, inflow, and the propulsion system. The study is conducted against UH-60 frequency responses identified from flight test data.

Mathematical Model

Background

The airframe and rotor model used in this study is a derivative of Sikorsky's Gen Hel simulation [10] and NASA/Ames real-time version [15] with several major modifications. These modifications are described in detail in Ref. [4]. Only a brief outline will be presented here. The model consists of a nonlinear, blade element representation of an articulated single main rotor helicopter with a rigid fuselage. The main rotor blades are individually formulated in the rotating frame as rigid bodies undergoing flap and lag motion. Elastic blade twist is accounted for empirically. A three-state dynamic inflow model [16] is used to account for the unsteadiness of the induced velocities at the main rotor disk. The calculations of tail rotor forces and moments are based on a the simplified, closed form Bailey solution [17], which employs actuator disc theory. The aerodynamic coefficients of all lifting surfaces, *i.e.* rotor blade section, fuselage body, and the empennage, are obtained from wind tunnel tests and implemented as table look-up in the model.

Propulsion system

The propulsion system model is derived for the General Electric T700-GE-700 turboshaft engine. The mathematical model used is based on the linear model of Ref. [12] which was extracted from a component-type simulation of Ref. [13], in which thermodynamic and kinematic models of the different engine components are combined to represent the dynamics of the whole engine. These models have been converted into a first order ordinary differential equation (ODE) form to be implemented with the UH-60 mathematical model used in this study.

The propulsion system model used in this study neglects the flexibility of the drive shaft and the gearbox, and therefore the equation of motion for the rotor speed degree of freedom is given simply by:

$$J_{tot}\dot{\Omega} = J_z \dot{r}_s + Q_e - Q_{req} \tag{1}$$

The product $J_z \dot{r}_s$ is an equivalent torque term which accounts for the inertial effects of the body's yaw acceleration $\dot{r}_s$ on the rotating components aligned along the vertical axis. This needs to be included because the gearbox output speed is defined with respect to the body-fixed axes.

The fuel control dynamics is represented by one equation that relates the fuel flow state W_1 to the rotor speed error $\Delta\Omega$, to the derivative and integral of the rotor speed error, $\Delta\dot{\Omega}$ and $\Delta\psi$, and to the time constant T_F:

$$T_F \dot{W}_1 + W_1 = K_D \Delta\dot{\Omega} + K_P \Delta\Omega + K_I \Delta\psi \tag{2}$$

Total fuel flow W_f also depends proportionally to the movement of the collective stick $\Delta\delta_{col}$ by an anticipation gain K_C. The expression for the total fuel flow is given by:

$$W_f = W_1 + K_C \Delta\delta_{col} \tag{3}$$

The remaining equations of the propulsion system include one for the gas generator speed N_G, and three for thermodynamic states, P_3, P_{41}, and P_{45}. They are given by:

$$\begin{aligned}
\dot{N}_G &= \frac{60}{2\pi}\frac{Q_{GT} - Q_C}{J_{GT}} && (4)\\
\dot{P}_3 &= K_{V_3} T_3 \left(W_{A_3} - W_{B_3} - W_{A_{31}}\right) && (5)\\
\dot{P}_{41} &= K_{V_{41}} T_{41} \left(W_{A_{31}} + W_f - W_{41}\right) && (6)\\
\dot{P}_{45} &= K_{V_{45}} T_{45} \left(W_{41} - W_{45} + B_3 K_{bl} W_{A_2}\right) && (7)
\end{aligned}$$

A complete derivation of these equations is given in Ref. [13].

Linearization

The equations describing the motion of fuselage and the rotor are fully coupled and they are formulated as set of first-order nonlinear ordinary differential equations such that the solution of the equations is conducted simultaneously. Linearization of the set of equations is conducted using finite difference approximations about a trim equilibrium point which is obtained using a coupled rotor and fuselage trim procedure. The trim procedure is valid for steady-level flights, climbs, descents, and coordinated turns, and is outlined in detail in Ref. [5].

In summary, the trim procedure is broken up into two phases. The first phase is based on a simultaneous solution of a set of nonlinear algebraic equations that satisfies the equilibrium requirements in an average sense. The second phase is based on a shooting method and it enforces the periodicity of all the states. Although this step may not be important for performance calculations, it becomes important during linearization, because even at small forward speeds, the steady state response of the aircraft is periodic and the linear model should be averaged over one rotor revolution.

The equilibrium requirements of the propulsion system add additional unknowns to the trim procedure. In addition to the equilibrium requirements for the rotor and the fuselage, the states of the propulsion system have to reach steady state value for a given flight condition. The initial requirement is the matching of the torque produced by the engine with the torque required by the aircraft. Since engine torque is a function of fuel flow W_f, the gas generator speed N_G, and the thermodynamic states, P_3, P_{41}, and P_{45}, these variables are varied until the following requirement is satisfied:

$$\int_0^{2\pi} \dot{\Omega} d\psi = 0 \tag{8}$$

On the completion of torque matching, the engine states are allowed to stabilize using an Euler step integration in the following way:

1. Update fuel flow W_f and the engine states N_G, P_3, P_{41}, and P_{45}:

$$\begin{aligned} W_{f_{k+1}} &= W_{f_k} + \dot{W}_{f_k}\,\Delta t && (9)\\ N_{G_{k+1}} &= N_{G_k} + \dot{N}_{G_k}\,\Delta t && (10)\\ P_{3_{k+1}} &= P_{3_k} + \dot{P}_{3_k}\,\Delta t && (11)\\ P_{41_{k+1}} &= P_{41_k} + \dot{P}_{41_k}\,\Delta t && (12)\\ P_{45_{k+1}} &= P_{45_k} + \dot{P}_{45_k}\,\Delta t && (13) \end{aligned}$$

   Δt is chosen as 0.0002 seconds and the derivative of fuel flow $\dot{W}_f$ is approximated as:

$$\dot{W}_f = 0.12(Q_{req} - Q_e) \tag{14}$$

   The initial values are obtained in the first phase of the trim procedure.

2. Compute the derivatives of N_G, P_3, P_{41}, and P_{45}:

$$\frac{d}{dt}\begin{bmatrix} N_G \\ P_3 \\ P_{41} \\ P_{45} \end{bmatrix} = \mathbf{f}(\mathbf{x}, \mathbf{u}; t) \tag{15}$$

3. Convergence of the engine states is achieved when the torque available matches the torque required $Q_{req} - Q_e$, and the derivatives of N_G, P_3, P_{41}, and P_{45} have reduced to zero. Otherwise, the procedure is repeated.

The final linear model is given by the following:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \end{aligned} \tag{16}$$

where

$$\begin{aligned} \mathbf{x} = \lfloor & u\ v\ w\ p\ q\ r\ \Phi\ \Theta\ \Psi\ \dot{\beta}_0\ \dot{\beta}_{1c}\ \dot{\beta}_{1s}\ \dot{\beta}_2 \\ & \beta_0\ \beta_{1c}\ \beta_{1s}\ \beta_2\ \dot{\zeta}_0\ \dot{\zeta}_{1c}\ \dot{\zeta}_{1s}\ \dot{\zeta}_2\ \zeta_0\ \zeta_{1c}\ \zeta_{1s}\ \zeta_2 \\ & \phi_{DYN}\ \dot{\phi}_{DYN}\ \nu_0\ \nu_{1c}\ \nu_{1s}\ \nu_t\ \nu_x\ \nu_y \\ & \Delta\psi\ \Omega\ N_G\ W_1\ P_3\ P_{41}\ P_{45}\rfloor^T \end{aligned} \tag{17}$$

and

$$\mathbf{u} = \lfloor \delta_{lat}\ \delta_{lon}\ \delta_{col}\ \delta_{ped}\ \delta_{stab} \rfloor^T \tag{18}$$

Numerical integration

When the dynamics of the propulsion system was included in the time history simulation, the interaction of the *fast* dynamics of the pressure states with the *slower* kinematic states (*i.e.* gas generator speed, power turbine speed) posed problems for the ODE solver because of the system's *stiff* nature. In Ref. [12], Euler integration with a very small time step was used at the expense of both high-order accuracy and computational speed. Ref. [13] applied a completely different approach. The pressure states were assumed to be fast enough such that they were considered to be instantaneous. Consequently, the differential equations for the pressure dynamics were transformed into nonlinear algebraic equations and the solutions for the states were computed iteratively. For this study, the integration is conducted using Gear's integration method for stiff systems [18]. The technique applies backward differentiation formula of order up to five and is especially applicable because it allows the simultaneous integration of all the states including the pressure states without having to use iteration loops, convergence criteria, and without having to reformulate the equations of motion.

Fuel Control Identification

A correct representation of rotor and propulsion system dynamic interaction requires a closed-loop model in which the fuel controller adjusts fuel flow continually to maintain the rotor at the desired speed. The electronic control unit (ECU) of a T700 was designed for this purpose, and a mathematical model with considerable amount of detail was developed in Ref. [13]. For this study, it was determined that a comparatively simpler PID model of a controller could adequately represent the dynamics of the controller in the frequency of interest for handling qualities and control system designs. The underlying assumption is that the complex dynamics of the ECU which includes numerous time delays, feedback loops, and hysteresis effects, can be lumped into a PID model of the controller. The frequency response system identification methods and the CIFER program discussed in Ref. [19] are used to determine the gains and the time constant of the PID controller.

Two sets of frequency responses were used to identify the fuel control system - rotor speed error to collective ($\Delta\Omega/\Delta\delta_{col}$) and fuel flow to collective ($W_f/\Delta\delta_{col}$). These are generated from time histories of the rotor speed and fuel flow variations. The pilot input is a collective stick frequency sweep, ranging from 0.3 rad/sec to about 30 rad/sec. The aircraft is forced to start and end in trim. A fast-Fourier transform using chirp z-transforms was conducted to yield the desired frequency responses and the coherence function, which measures the linearity between the input and the output, was used as a primary indicator of the accuracy of the identification. A detailed discussion of the properties of the coherence function and the methods of chirp z-transforms, as they pertain to the present study, can be found in Ref. [20].

The model structure used for the identification is illustrated in Fig. 1. The items modeled are collective anticipation K_C, proportional gain K_P, derivative gain K_D, integral gain K_I, and a filter with a time constant T_F. The plant is the linear model extracted from the mathematical model presented in this study (see Eq. (16)).

The undetermined coefficients of the fuel controller are then modeled as free parameters in the state space formulation of Eq. (16) in the following way:

$$\begin{bmatrix} 1 & & & & & \\ & & & & & \\ & 1 & 0 & 0 & 0 & 0 \\ & 0 & 1 & 0 & 0 & 0 \\ & 0 & -K_D & T_F & 0 & 0 \\ & 0 & 0 & 0 & 1 & 0 \\ & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \vdots \\ \Delta\dot{v} \\ \Delta\dot{\Omega} \\ \dot{W}_1 \\ \dot{W}_2 \\ \dot{\delta}_{col} \end{bmatrix} =$$

$$\begin{bmatrix} a_{11} & & & & & b_{13} \\ & & & & & \vdots \\ & 0 & 1 & 0 & 0 & 0 \\ & \cdot & \cdot & \cdot & \cdot & b_{\Delta\Omega 3} \\ & K_I & K_P & -1 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ \vdots \\ \Delta v \\ \Delta\Omega \\ W_1 \\ W_2 \\ \delta_{col} \end{bmatrix}$$

$$+ \begin{bmatrix} b_{1.1} & b_{1.2} & 0 & b_{1.4} & b_{1.5} \\ & & & & \vdots \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_C & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_{lat} \\ \delta_{lon} \\ \dot{\delta}_{col} \\ \delta_{ped} \\ \delta_{stab} \end{bmatrix} \qquad (19)$$

The two outputs, fuel flow W_f and the rotor speed error $\Delta\Omega$, are then given by:

$$W_f = W_1 + W_2 \qquad (20)$$

$$\Delta\Omega = \Delta\Omega \qquad (21)$$

The optimization scheme in CIFER uses a secant search method to determine the free parameters by minimizing the error between the model and the frequency responses [19]. To emphasize the more accurate portions of data, fitting errors are weighted by coherence. The converged results are presented in Table 1 and Fig. 2. In the table, "CR-Bound" refers to the Cramer-Rao bound, which is a measure of correlation among parameters. It is given as a percentage of the parameter value. The insensitivity of the parameters to the optimization scheme, also measured in percent, is labeled "Insen". Parameters with Cramer-Rao bounds less than 20% and insensitivities less than 10% are considered to have acceptable accuracy [19]. The cost function is also shown for each of the frequency responses identified. It is an indicator of the accuracy of parameter identification and a cost of less than 100 is considered accurate. The derivative gain K_D is not listed in Table 1 because it was found to be insensitive and highly correlated with the collective anticipation gain K_C. It was eliminated from the model structure.

Parameter	Value	CR bound	Insens
T_F	0.670E-01	19.8	6.82
K_I	-0.909E-01	8.18	3.53
K_P	-0.595E-01	5.81	2.16
K_C	0.150E-02	30.8	15.0

Table 1: Final results from fuel control ID.

To verify the accuracy of the identified model, time histories of the rotor speed Ω and the fuel flow W_f to a one-inch down collective step input were generated with the linear model and compared against flight test responses. They are shown in Fig. 3 along with the primary collective stick input "col" and the three off-axis inputs - lateral "lat", longitudinal "lon", and pedal "ped". The predictive capability of the model in the time domain is reasonable but reflects the rather high cost function during the identification of the rotor speed error to collective response $\Delta\Omega/\delta_{col}$ (cost=220.4). Achieving a more accurate fit was limited by three factors. First, the linear model of the airframe and the propulsion system is extracted from predictive simulation model and therefore is not entirely accurate; second, the quality of the data for frequencies below 2-3 rad/sec was rather poor; and third, the model structure chosen for the identification is a simplification of a more complex system.

Results

Helicopter configuration

In this section, the frequency response predicted from the high-order linear model is compared with that derived from flight tests of a UH-60A Black Hawk. The aircraft has a nominal gross weight of about 14000 lbs., fuselage station center of gravity position of 361.3 inches and waterline center of gravity position of 251.5 inches. It is flying at an altitude of 200 feet for hover and 1000 feet for forward flight. Flight tests have been conducted with frequency sweeps in each of the control axes and the frequency response plots are prepared using CIFER system identification techniques discussed earlier [19]. The stability augmentation system (SAS) was turned off for roll and yaw axes, but for safety reasons was switch on for the pitch axes. The flight path stabilization was turned off for all control sweep records. The actuator dynamics was modeled as a 50 msec time delay in each of the four control axes.

Propulsion system analysis

In this section, the effects of the propulsion system dynamics are assessed by comparing frequency responses of flight test data against a linear model with and without the dynamics of the propulsion system. Standard model-order reduction techniques are applied to the full-order model to generate the linear model without the dynamics of the propulsion system. The frequency responses are presented in Figs. 4 through 7. The flight test data are represented by a solid line, the full-order model by a dashed line, and the lower-order model by a dotted line. The state vector of the lower-order model is given by:

$$\begin{aligned}\mathbf{x} \;=\; & \lfloor u\ v\ w\ p\ q\ r\ \Phi\ \Theta\ \Psi\ \dot{\beta}_0\ \dot{\beta}_{1c}\ \dot{\beta}_{1s}\ \dot{\beta}_2 \\ & \beta_0\ \beta_{1c}\ \beta_{1s}\ \beta_2\ \dot{\zeta}_0\ \dot{\zeta}_{1c}\ \dot{\zeta}_{1s}\ \dot{\zeta}_2\ \zeta_0\ \zeta_{1c}\ \zeta_{1s}\ \zeta_2 \\ & \phi_{DYN}\ \dot{\phi}_{DYN}\ \nu_0\ \nu_{1c}\ \nu_{1s}\ \nu_t\ \nu_x\ \nu_y\rfloor^T \qquad (22)\end{aligned}$$

As seen in Figs. 4 and 5, the dynamics of the propulsion system affects very little the predictions of the frequency responses to a lateral or longitudinal cyclic inputs. The more significant effects are observed in the aircraft responses to a collective and pedal inputs, Figs. 6 and 7 respectively. This is not surprising since the torque variations in a rotorcraft occur normally during maneuvers in the heave and the yaw axes. In the vertical acceleration response to a collective input, indicated as "a_z/col", the magnitude prediction is attenuated by 1-2 dB and the phase lag decreased by about 10 degrees as a result of including the propulsion system dynamics. The effects are observed mainly in the mid-frequency range, from 1 rad/sec to about 10 rad/sec. They exist mainly because in the lower-order model, the rotor speed is constant, *i.e.* perfect governor. In the full-order model, the dynamic interaction between the rotor speed and the engine governor is explicitly modeled and causes the magnitude of the vertical acceleration response to be attenuated. The difference is also observed in the yaw rate response to a collective input "r/col." Here, the differences are more significant with up to 10 dB's in the magnitude response and 100 degrees in the phase response. Likewise, in Fig. 7, the yaw rate response to a pedal input "r/ped" shows up to a 5 dB improvement in the magnitude response and up to a 30 degree improvement in the phase response.

On-axis frequency response

Figs. 4 through 7 contain results of both the on-axis and cross-coupled frequency responses validation study. This section will examine the quality of the validation of the on-axis responses. When comparing the responses, it is very important to assess their accuracy by examining the coherence curves. As a general rule of thumb, coherence is usually poor at both ends of the frequency spectrum due to decreased pilot inputs. The reduced pilot inputs lead to an increase of noise to signal ratios that corrupts the quality of the frequency responses.

The comparison for the roll rate response to a lateral cyclic input is indicated as "p/lat" in Fig. 4 and pitch rate response to a longitudinal input is indicated as "q/lon" in Fig. 5. For both sets of responses, the correlation is good to excellent for frequencies above 1 rad/sec. Both were successful in capturing the notch type response associated with the regressive lag mode at about 20 rad/sec. At low frequencies, the agreement of the predicted responses are quite poor, but here, the low coherence reflects poor confidence in the flight identified results.

The hover vertical acceleration response to collective stick input "a_z/col" is shown in Fig. 6. The quality of the analytical model prediction is good to excellent for frequencies above 0.5 rad/sec. The model is successful in capturing the magnitude rise associated with inflow dynamics beginning at about 3 rad/sec. However, a rather large discrepancy exists in the phase response below 0.5 rad/sec. The error is associated with an overprediction of the heave damping Z_w. It is common for momentum theory to predict values of Z_w that are up to three times larger than what is identified from flight data [21]. Houston, in Ref. [22] attributes this problem to the nonuniformity of the inflow environment and suggests an empirical factor to correct the uniform inflow assumption.

Amplitude and phase of the yaw rate response to pedal input, indicated as "r/ped" in Fig. 7 is in good agreement with flight tests along the complete frequency range, with the exception of the magnitude response near 2 rad/sec. The model overpredicts the magnitude by as much as 5 dB. A possible source of improvement may be in switching from an actuator disc model of the tail rotor to a more refined blade element type of representation.

Off-axis frequency response

The off-axis responses predicted using the high-order linear model of this study are presented alongside the on-axis responses in Figs. 4 through 7. The accuracy of the model predictions in the off-axis is important when designing control laws to decouple the cross-coupled dynamics. The cross-coupling effect regarded as very important is that of pitch and roll. The responses are indicated as "q/lat" and "p/lon" in Figs. 4 and 5 respectively and exhibit very poor correlation with flight test data. The problem of predicting the correct pitch/roll coupling have been observed for quite some time, more specifically, in the AH-64 simulation of Ref. [23], BO-105 simulation of Ref. [24], and a UH-60 simulation of Ref. [25], and is often attributed to the complex interaction of the fuselage and the main rotor wake.

The predictions of vertical acceleration off-axis response are likewise poor. However, these predictions were limited by low levels of excitation in the off-axis resulting in low values of the coherence function. Nevertheless, each of the off-axis response of the vertical acceleration, "a_z/lat" in Fig. 4, "a_z/lon" in Fig. 5, and "a_z/ped" in Fig. 7, shows poor correlation with flight test data for both the magnitude and the phase. The off-axis response with the best overall correlation is the yaw rate response. In Fig. 4, the prediction of the yaw rate response to a lateral stick input "r/lat" is good for most of the frequency range, while the yaw rate response prediction to a longitudinal stick input, "r/lon" in Fig. 5, is less accurate. And as discussed earlier, the off-axis response of the yaw rate to a collective input, "r/col" of Fig. 6, is much improved when the propulsion system dynamics are included in the analysis.

Forward flight frequency response

Figs. 8 and 9 contain the on-axis frequency responses of the UH-60 flying at 80 knots and 120 knots respectively. No off-axis coupled responses are shown. The fuel controller is assumed to remain relatively constant with forward speed and therefore the gains used are identical to those identified for hover. In the roll rate response to a lateral stick input "p/lat", the results show that both amplitude and phase are predicted well at frequencies above 0.5 rad/sec. The notch type response and the phase increase associated with the regressive lag mode are captured by the model, although the frequency of the mode is slightly underpredicted. The same general observations hold true for the V = 120 knots case, "p/lat" in Figure 9, except now the frequency of the lag regressive mode is predicted correctly.

The amplitudes of the pitch rate response to a longitudinal input, "q/lon", are predicted accurately at both speeds for frequencies above 0.7 rad/sec. Below 0.7 rad/sec, the amplitude is slightly underpredicted for 120 knots. For 80 knots, as with the roll rate case, the prediction remains reasonably accurate even at the lower frequencies. On the other hand, a boost in phase is predicted at both speeds at low frequencies. Above 2 rad/sec, the prediction is accurate at both speeds. The agreement remains good up to 20-30 rad/sec.

For the vertical acceleration response to a collective input "a_z/col", the agreement is reasonable in the frequency range where the coherence is good, 1-20 rad/sec for 80 knots and 4-20 rad/sec for 120 knots. For both speeds, the magnitude is overpredicted consistently through out the entire frequency spectrum and the correlation worsens as frequency is reduced. It may be possible that the relatively crude modeling of rotor blade torsion can account for some of the low frequency errors.

Likewise, the yaw rate responses to pedal input "r/ped", is accurate over a rather limited range of frequency, ranging from about 1 to 10-12 rad/sec. In this frequency range, the agreement with flight test data for both the magnitude and phase portion is reasonable. The linear model overpredicts the amplitude by about 3 dB at both speeds. This may be related to the similar overprediction in the vertical axis because the canted tail rotor couples the two degrees of freedom.

Conclusions

A nonlinear mathematical model of helicopter flight dynamics that can describe the dynamics of the fuselage, rotor, and of the propulsion system has been formulated. A high-order linear, constant coefficient set of equations of motion has been extracted from this nonlinear model via numerical perturbation. These equations describe the small perturbation dynamics about an equilibrium position. A validation study of the mathematical model was conducted in the frequency domain through comparisons of the frequency responses predicted by the linearized model with flight test data. The study was conducted for the UH-60 Black Hawk in hover and flying at forward speeds of 80 and 120 knots.

In addition, a coupled rotor-fuselage-engine trim procedure has been developed to obtain the equilibrium position about which the mathematical model is linearized. The procedure calculates the steady state response of the rotor blades and engine states, as well as the trim attitudes and rates of the fuselage. The procedure retains the periodicity of the rotor, fuselage, and engine states and is valid for both straight flight and steady coordinated turns.

Based on the results presented in this study, the following conclusions may be drawn:

1. State space parameter identification techniques were used successfully to identify a model of the fuel controller using a combination of frequency responses obtained from flight tests and a linear model of the plant extracted from a high-order linear model of a helicopter.

2. The effects of the propulsion system dynamics are observed to be mainly a mid-frequency phenomena, between 1-10 rad/sec. The effect was most predominant in the heave and yaw degrees of freedom and most significant in the yaw rate response to a collective input.

3. The frequency domain correlation study shows that the predicted on-axis frequency responses are in good to excellent agreement with the flight test. On the other hand, the off-axis responses were often predicted inaccurately.

References

[1] *Airworthiness Design Standard, Handling Qualities, Rotary Wing*, U.S. Army AVSCOM ADS-33, May 1987.

[2] Hall. W.E.. Jr.. and Bryson. A.E.. Jr.. "Inclusion of Rotor Dynamics in Controller Design." *Journal of Aircraft*. Vol. 10. No. 4. 1973. pp. 200-206.

[3] Chen. R.T.N.. and Hindson. W.S.. "Influence of High-Order Dynamics on Helicopter Flight-Control Bandwidth." *Journal of Guidance. Control. and Dynamics*. Vol. 9. No. 2. 1986. pp. 190-197.

[4] Kim. F.D.. Celi. R.. and Tischler. M.B.. "High-Order State Space Models of Helicopter Flight Mechanics," *Proceedings of the Sixteenth European Rotorcraft Forum*. Glasgow. Scotland. Sept. 1990.

[5] Kim. F.D.. Celi. R.. and Tischler. M.B.. "Forward Flight Trim Calculation and Frequency Response Validation of a High-Order Helicopter Simulation Model." *Proceedings of the 47th Annual Forum of the American Helicopter Society*. Phoenix. Arizona. May 1991.

[6] Ingle. S.J., and Celi. R.. "Effects of Higher Order Dynamics on Helicopter Flight Control Law Design." *Proceedings of the 48th Annual Forum of the American Helicopter Society*. Washington, D.C.. June 1992.

[7] Sanders. J.C.. "Influence of Rotor-Engine Torsional Oscillation on Control of Gas-Turbine Engine Geared to Helicopter Rotor." NACA TN-3027, Oct. 1953.

[8] Fredrickson. C.. Rumford, K.. and Stephenson. C.. "Factors Affecting Fuel Control Stability of a Turbine Engine/Helicopter Rotor Drive System." *Journal of the American Helicopter Society*, Vol. 17. No. 1, Jan. 1972.

[9] Kuczynski, W.A., Cooper. D.E.. Twomey. W.J., and Howlett, J.J., "The Influence of Engine/Fuel Control Design on Helicopter Dynamics and Handling Qualities," *Journal of the American Helicopter Society*, Vol. 25, No. 2, Apr. 1980.

[10] Howlett. J.J.. "UH-60A Black Hawk Engineering Simulation Program - Volume II - Background Report." NASA CR-166310. Dec. 1981.

[11] Hull. R.. "Development of a Rotorcraft/Propulsion Dynamic Interface Analysis: Volume I." NASA CR-166380. Aug. 1982.

[12] Ockier. C.J.. and Celi. R.. "Dynamics and Aeroelasticity of a Coupled Rotor-Propulsion System in Hover." Paper AIAA-91-1220-CP, *Proceedings of the AIAA/ASME/ASCE/AHS 32nd Structures. Structural Dynamics and Materials Conference*. Baltimore, Maryland, Apr. 1991, Part 3, pp. 2058-2070.

[13] Ballin, M.G., "A High Fidelity Real-Time Simulation of a Small Turboshaft Engine," NASA TM-100991, July 1988.

[14] Kaplita. T.T.. Driscoll. J.T.. Diftler. M.A.. and Wong. S.W., "Helicopter Simulation Development by Correlation With Frequency Sweep Flight Test Data." *Proceedings of the 45th Annual Forum of the American Helicopter Society*. Boston. Massachusetts. May 1989. pp. 681-692.

[15] Ballin. M.G.. "Validation of a Real-Time Engineering Simulation of the UH-60A Helicopter." NASA TM-88360. Feb. 1987.

[16] Pitt. D.M.. and Peters. D.A.. "Theoretical Prediction of Dynamic Inflow Derivatives." *Vertica*. Vol. 5. No. 1. 1981. pp. 21-34.

[17] Bailey. F.J., Jr.. "A Simplified Theoretical Method of Determining the Characteristics of a Lifting Rotor in Forward Flight." NACA Report 716. 1941.

[18] Gear. G.W.. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall. Englewood Cliffs. New Jersey. 1971.

[19] Tischler. M.B., and Cauffman, M.G., "Frequency-Response Method for Rotorcraft System Identification with Applications to the BO-105 Helicopter." *Proceedings of the 46th Annual Forum of the American Helicopter Society*, Washington. D.C.. May 1990.

[20] Tischler, M.B.. "Frequency Response Identification of XV-15 Tilt-Rotor Aircraft Dynamics." NASA TM-89428. May 1987.

[21] Tischler. M.B.. Fletcher. J.W.. Diekmann. V.L.. Williams. R.A.. and Cason. R.W.. "Demonstration of Frequency-Sweep Testing Technique Using a Bell 214-ST Helicopter." NASA TM-89422. Apr. 1987.

[22] Houston. S.S. "Identification of a Coupled Body/Coning/Inflow Model of Puma Vertical Response in Hover." *Vertica*. Vol. 13. No. 3. 1989. pp. 229-249.

[23] Chaimovich, M.. Mansur. M.H.. Rosen. A., Tischler. M.B.. and Rand. O.. "Investigation of the Flight Mechanics Simulation of a Hovering Helicopter." *Proceedings of the 48th Annual Forum of the American Helicopter Society*, Washington, D.C.. June 1992.

[24] Kaletka. J.. "Rotorcraft System Identification - Case Study II: BO-105." AGARD-AR-280. Sep. 1991. pp. 133-175.

[25] Takahashi. M.D.. "A Flight Dynamic Helicopter Mathematical Model with a Single Flap-Lag-Torsion Main Rotor." NASA TM-102267. Feb. 1990.

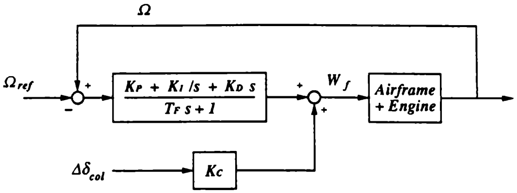


Figure 1: Model structure used for fuel control ID.

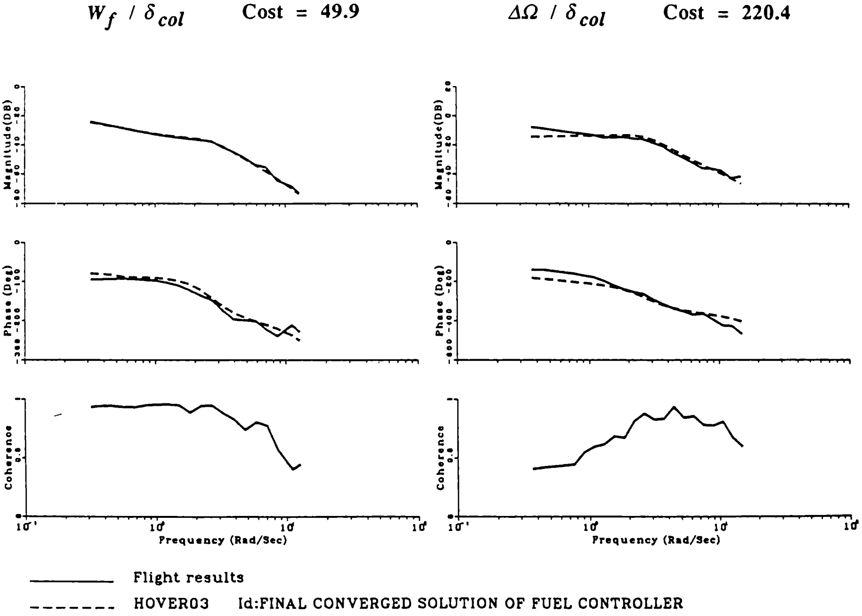


Figure 2: Final converged results of fuel control ID.

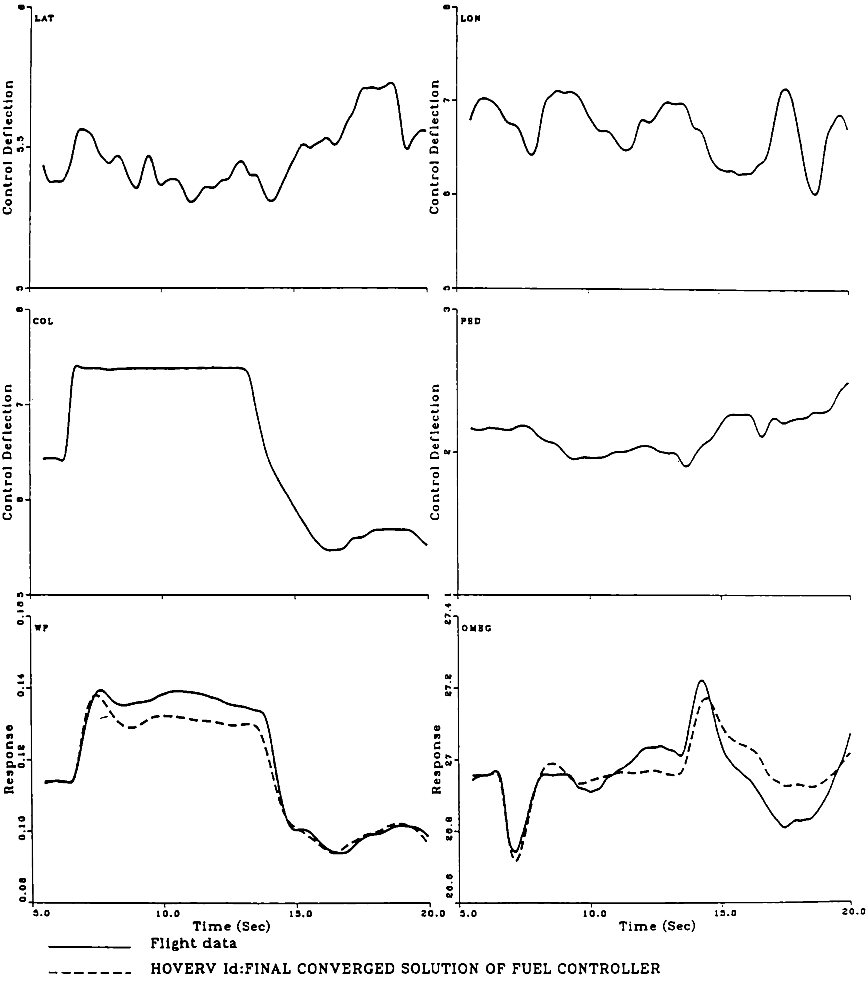


Figure 3: Time history comparisons to a 1-inch collective step input.

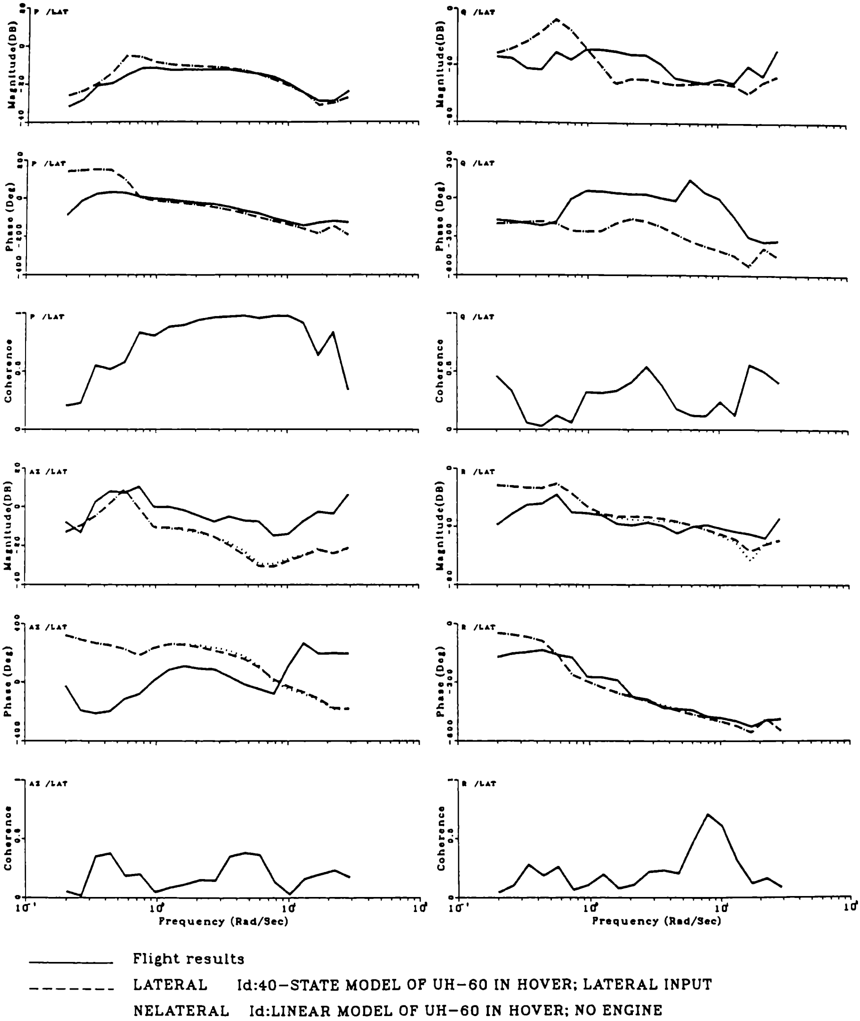


Figure 4: Responses of the UH-60 in hover to a lateral input δ_{lat}.

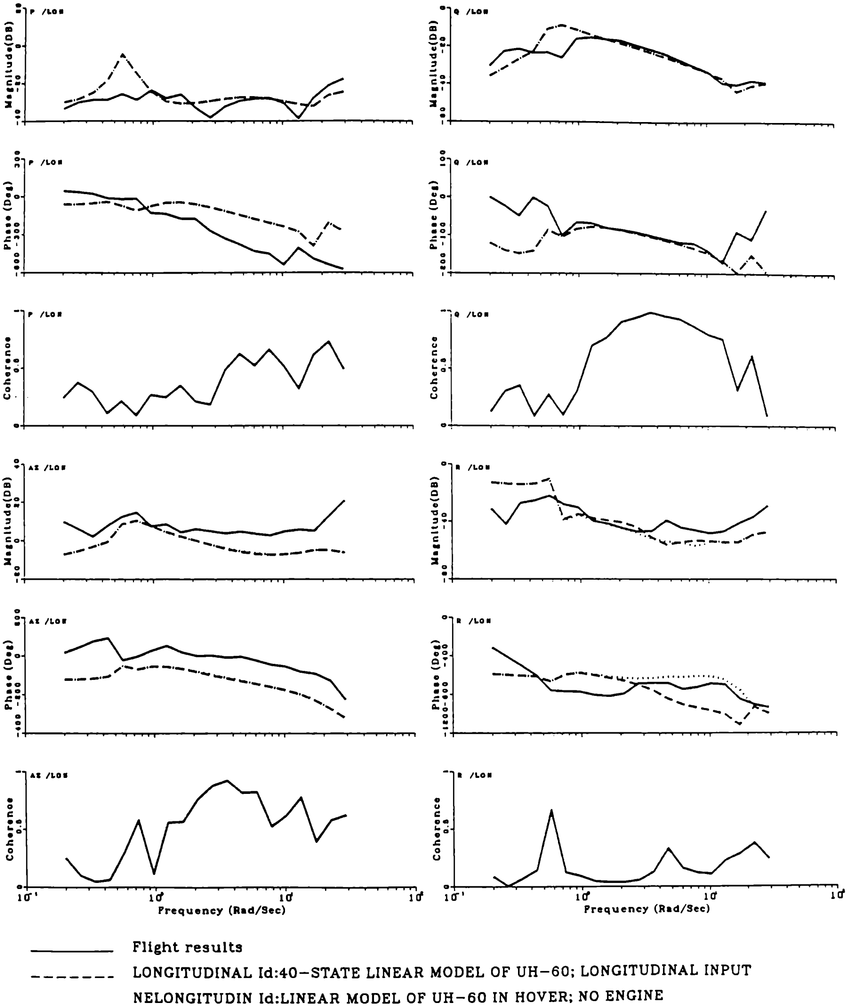


Figure 5: Responses of the UH-60 in hover to a longitudinal input δ_{lon}.

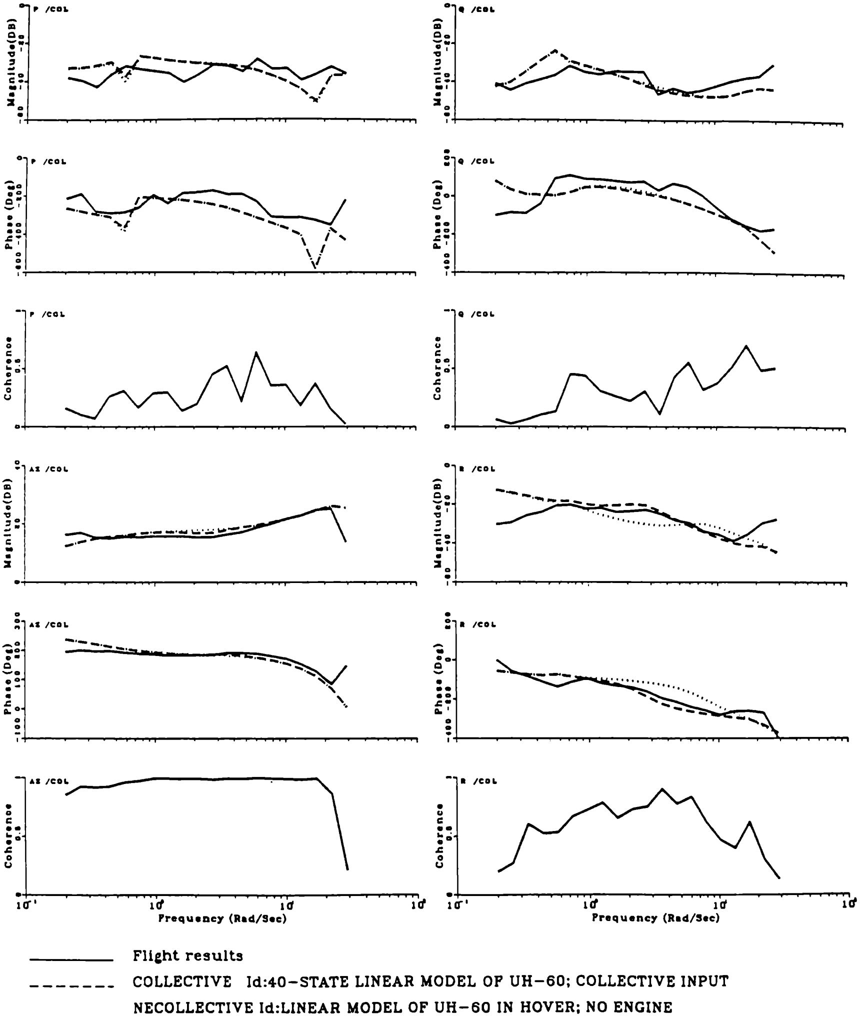


Figure 6: Responses of the UH-60 in hover to a collective input δ_{col}.

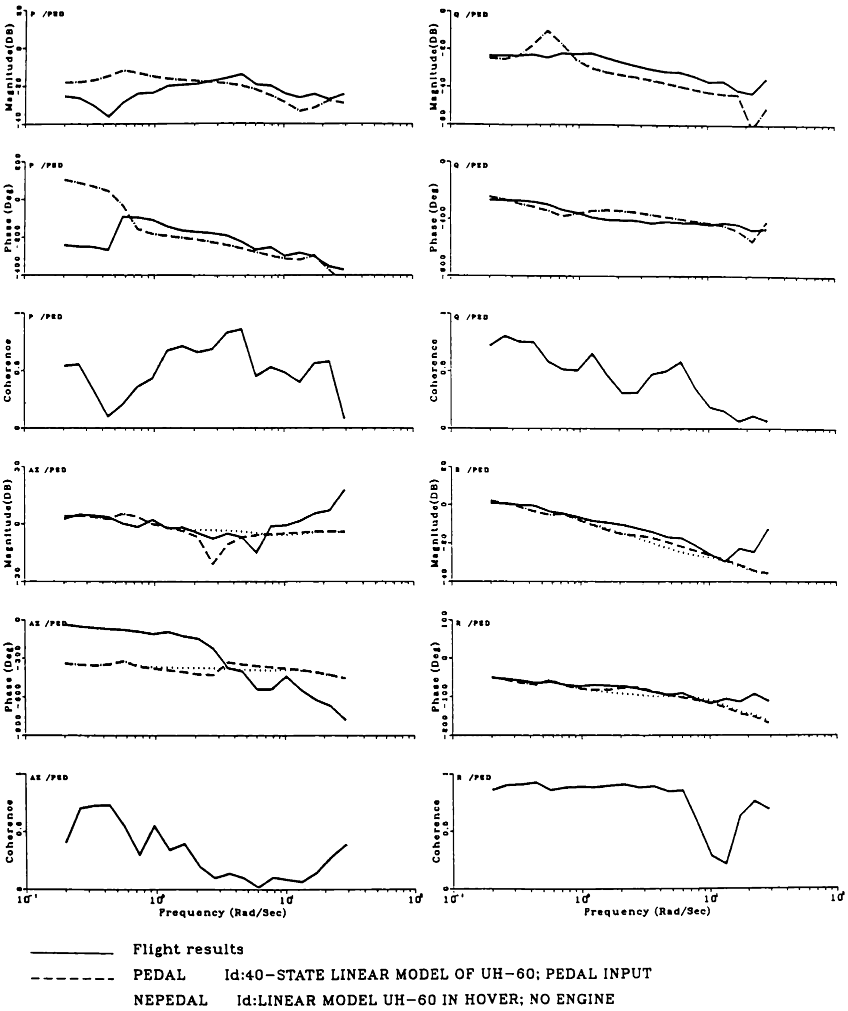


Figure 7: Responses of the UH-60 in hover to a pedal input δ_{ped}.

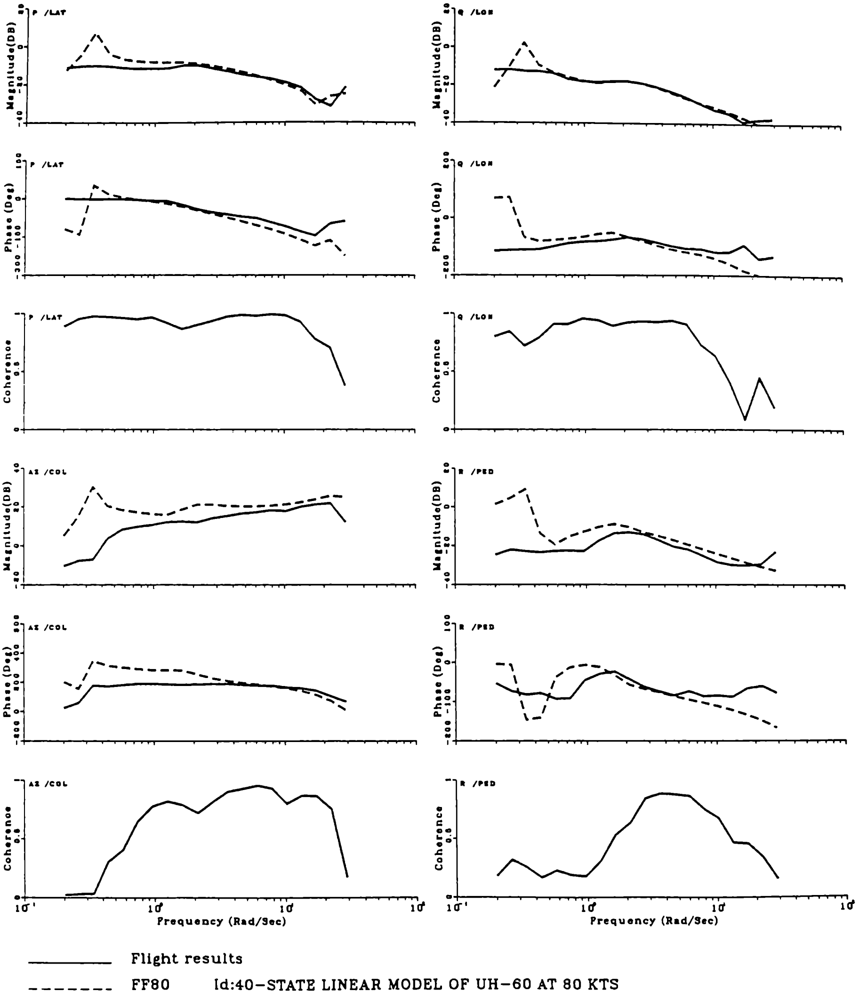


Figure 8: On-axis responses of the UH-60 flying at 80 knots.

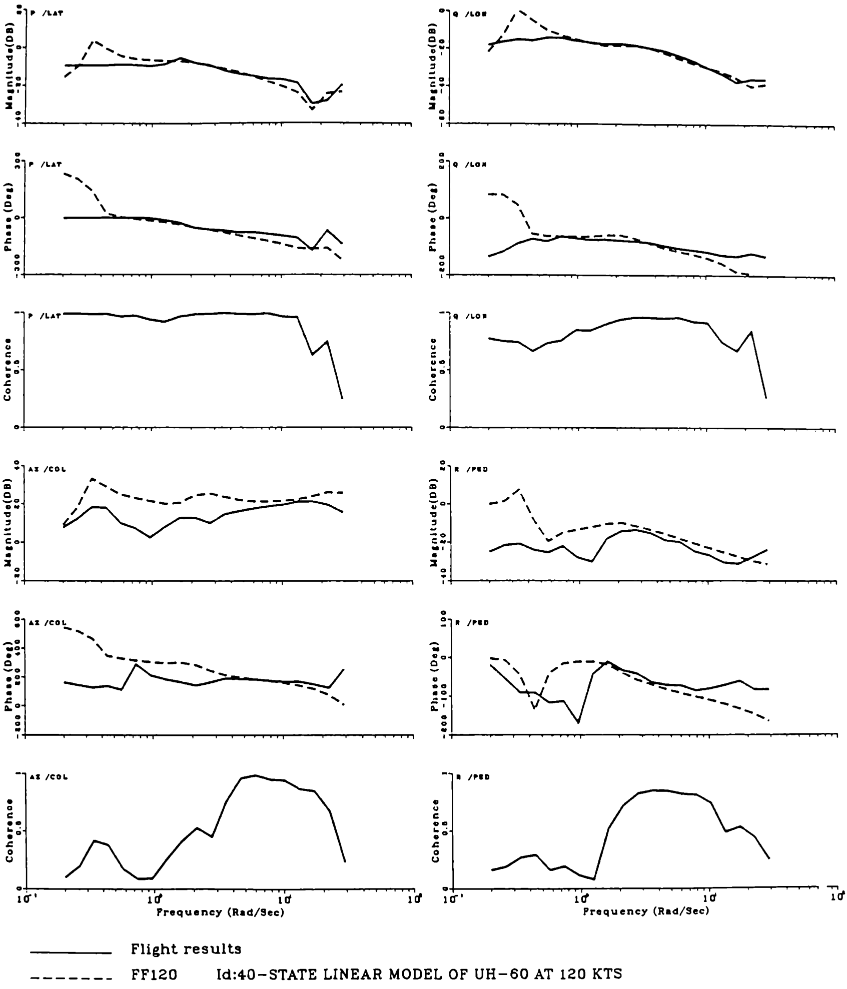


Figure 9: On-axis responses of the UH-60 flying at 120 knots.

AIAA-92-4151-CP

GENERIC HELICOPTER FOR MULTISHIP SIMULATIONS

Larry A. Moody
Technical Specialist
McDonnell Aircraft Company

Abstract

A new software model was created to fill the need for a generic helicopter to be used for enhanced environment realism in multiship combat simulations. Since the purpose of the generic helicopter model is to allow evaluation of weapons systems, human factors issues, or other issues not directly related to the helicopter, the main objective of the new model was to execute with a minimum of computational processing while maintaining acceptable performance and handling characteristics. This objective was met by taking a different and innovative approach. Rather than create a low fidelity helicopter with minimum control laws, a set of high fidelity control laws were used with a minimal aerodynamics reaction model. This approach yielded a helicopter model that not only achieved the objectives of low processing demand and acceptable performance and handling characteristics, but also retained all of the control system's pilot relief modes, such as altitude and heading hold modes.

Nomenclature

L	lift force, lbs.
D	drag force, lbs.
ρ	atmospheric density, slugs/ft^3
c	chord, ft.
r	blade radius, ft.
V	velocity, ft/sec.
θ	blade angle, degrees
α	angle of attack, degrees
a	lift curve slope, nondimensional
U	relative velocity, ft/sec.
Ω	angular velocity, radians/sec.
C_{d0}	drag coeff. at $\alpha = 0$, nondimensional
b	number of blades
υ	induced velocity, ft./sec.

Background

A recent multiship combat simulation effort at MCAIR required a helicopter model for additional environment realism. The simulation was hosted on a system of several distributed microprocessors. This concept allowed a large number of separate CPUs to process simultaneously; however, the speed of any one processor was limited.

The only previous helicopter simulation done at MCAIR was the preliminary LHX simulation in 1984-1985. This effort was a high fidelity single ship simulation. Pilot control was provided through a single four-axis, side-mounted controller rather than the conventional collective, cyclic, and rudder pedals.

This new controller approach simplified the incorporation of several control system modes, such as altitude hold, turn rate hold, and certain other mode changes that enabled the helicopter to be controlled like an airplane at high speeds. The control system used for this effort was the ARTI* control system. The helicopter math model originated at Hughes Helicopters (now MDHC) and was modified for high frequency dynamics by Advanced Rotocraft Technology in Mountain View, CA.

When rehosted on the microsystem computers, the high resolution model ran in 100 milliseconds even though the desired frame was 33 milliseconds. Since this model is highly coupled in all axes, it is not possible to break the model into subsections and coprocess the different axes simultaneously to achieve the desired reduction in computer time required. Furthermore, the high degree of resolution of this model was not required for this simulation. It was only necessary to have reasonable helicopter performance and handling characteristics, and still process the model within the allotted frame time.

Approach

To satisfy the computational and performance requirements, it was necessary to create an entirely new, generic helicopter model. To maintain similar handling characteristics as the high fidelity model, the approach taken was to maintain the ARTI control system and build a basic aerodynamics model to interface with the control system. This approach ensured that altitude hold, turn rate hold, and other control modes were maintained for ease of control by the simulation personnel operating the helicopter model.

The generic helicopter model replaces the main rotor, tail rotor, horizontal tail, vertical tail, and fuselage logic of the high fidelity model with the logic summarized in this report. The logic to derive the aerodynamics model begins with basic blade element theory. Referring to Fig. 1, it can be seen that the basic lift equation for a differential blade section dr, and the corresponding relationships for relative and tangential velocity are

$$dL = (1/2)\rho(c\ dr)V^2(\alpha\ a)$$

$$U = \mathrm{sqrt}(\ U_t^2 + U_p^2\ )$$

$$U_t = \Omega\ r$$

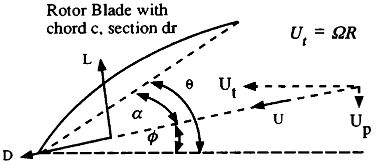


Figure 1. Blade Cross Section

Since the blade angle, θ, is known (it is simply the swashplate actuator angle plus the blade twist angle), it is desirable to replace α with a relationship in terms of θ. Using the small angle approximation,

$$\theta = \alpha + \varphi$$

$$\varphi = \tan(U_p / U_t) = U_p / U_t$$

$$\alpha = \theta - U_p / U_t$$

then the differential lift relationship can be written as:

$$dL = (1/2)\ \rho ac\ dr\ (U_t^2 + U_p^2)(\theta - U_p / U_t)$$

A similar approach, and assumption of constant drag coefficient, yield the relationship for drag on a differential blade section:

$$dD = (1/2)\ \rho c\ dr\ C_{d0}\ (U_t^2 + U_p^2).$$

The relationship of total blade thrust and torque to blade section lift and drag can be derived by referring to Fig. 2,

$$F_1 = L \cos\varphi - D \sin\varphi$$

$$F_2 = L \sin\varphi - D \cos\varphi,$$

where the differential form is written as

$$dF_1 = (\ dL\ U_t - dD\ U_p\ ) / \mathrm{sqrt}(\ U_t^2 + U_p^2\ )$$

$$dF_2 = (\ dL\ U_p\ +\ dD\ U_t\ )\ /\ \mathrm{sqrt}(\ U_t^2 + U_p^2\ )$$

by making the small angle assumptions as follows:

$$\cos\varphi = U_t\ /\ \mathrm{sqrt}(\ U_t^2 + U_p^2)$$

$$\sin\varphi = U_p\ /\ \mathrm{sqrt}(\ U_t^2 + U_p^2).$$

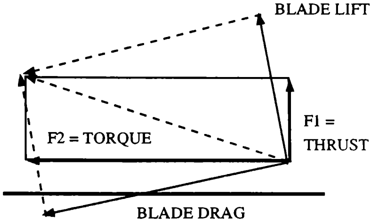


Figure 2. Blade Thrust and Torque

Now it is necessary to make several sweeping assumptions, governed by the requirements for a basic generic model that executes as fast as possible. First, assume that U_p^2 in the term $\mathrm{sqrt}(\ U_t^2 + U_p^2)$ is insignificant compared to the other term, and can be ignored. Furthermore, assume that the ratio C_{d0} / a is much less than 1, so the term of $(1 + C_{d0}/a)$ can reasonably be approximated by 1. Then substituting the derived values for dL and dD into the differential form of F_1 and F_2, rearranging and collecting terms, produces the following relationships:

$$dF_1 = (1/2)\ \rho ac\ dr\ [\ U_t^2\ \theta - U_pU_t\ ]$$

$$dF_2 = (1/2)\ \rho ac\ dr\ [\ U_tU_p\ \theta\ -\ U_p^2 + (\ C_{d0} / a\ )\ U_t^2\ ]$$

Integrating over the radius r, and multiplying by the total number of blades produces the total thrust and torque.

$$F_1 = (1/2)\ \rho\ a\ b\ c \int_0^R [\ U_t^2\ \theta - U_pU_t\ ]\ dr$$

$$F_2 = (1/2)\ \rho a\ b\ c \int_0^R [\ \Omega\ r(\ V + \upsilon\ )\ \theta - (\ V + \upsilon\ )^2 + \Omega^2\ r^2(\ C_{d0} / a\ )\ ]\ dr$$

The next logical step is to write the variables to be integrated in terms of r if they are functions of r, then integrate the total equation with respect to r. Since θ is the blade angle, it can be decomposed into the blade angle due to twist, θ_t, and the blade angle due to collective input, θ_c. The blade angle due to collective input, θ_c is a function of pilot inputs and control system outputs and will remain as an independent variable. The blade twist, θ_t, can be calculated as a function of blade radius if the assumption were made that lift along the blade was constant. So to calculate blade twist, start with the basic lift equation again, except hold lift L constant:

$$L = (1/2)\rho A V^2 C_l$$

Once again, assume linear aerodynamics, so that $C_l = a\,\alpha$

$$L = (1/2)\rho A V^2 a\,\alpha$$

Since lift L is constant, let $d = 2L / (\rho A a)$, and assuming constant altitude, then d = const.:

$$d = V^2\alpha$$

It can also be assumed that the relative velocity V is almost entirely due to the tangential velocity component $U_t = \Omega r$;

$$d = \Omega^2 r^2 \alpha$$

Note that the blade angular velocity Ω is constant for most flight conditions except for emergency autorotation and startup (rotor thrust is controlled by the blade angles, and the engine must increase or decrease speed to match the required torque). Since Ω is constant, then :

$$\alpha = d / \Omega^2 r^2 = \text{const} / r^2$$

The angle of attack required to maintain constant lift is the same as the blade twist angle, θ_t, so letting C_1 = const., the above equation can be rewritten in terms of the radius r, and a constant.

$$\theta_t = C_1 / r^2$$

The tangential velocity component U_t can be written in terms of r as Ω r. The perpendicular velocity component U_p can be written as ($V_z + \upsilon$), where V_z is the helicopter vertical velocity and υ is the vertical velocity induced by the flowfield of the rotors. With some assumptions, it is possible to write the induced velocity in terms of r. The calculation of induced velocity is based on momentum theory, which treats the rotor as an actuator disk, as displayed in Fig. 3, and deals with conservation of mass, volume, and energy of the total rotor system.

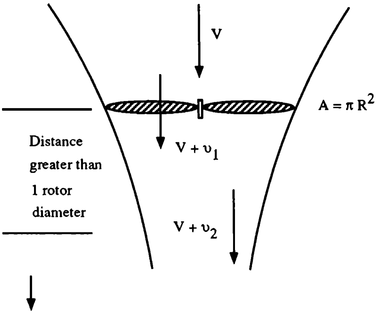


Figure 3. Calculating Induced Velocity

Using the momentum theory approach to equate the power requirements into the disk with the energy imparted to the flowfield by the disk shows that the induced velocity at the rotor disk is one half the value of the induced velocity at a distance greater than 1 rotor diameter away from the disk.

$$\upsilon_1 = \upsilon_2 / 2$$

Substituting the induced velocity relationship back into the force equation

$$F = 2\rho\pi R^2 (V + \upsilon_1)\upsilon_1$$

This force is the same as the total blade thrust calculated earlier. In hover V is zero, so assuming hover (or small vertical velocity conditions), then

$$T = 2\rho\pi R^2 \upsilon_1^2$$

$$\upsilon_1 = \text{sqrt}\,[\,T / (2\rho\pi R^2)\,]$$

The value of the induced velocity at the rotor υ_1 is the same as the induced velocity term in the total rotor thrust equation υ, so:

$$\upsilon = \text{sqrt}\,[\,T / (2\rho\pi R^2)\,]$$

Substituting the induced velocity and blade twist relationships back into the total thrust and torque equation, integrating with respect to radius, collecting the constants, eliminating insignificant terms, and rearranging, produces the following equations:

$$F_1 = \rho\ (C_2 \theta_c + C_3 - C_4\ V_z) \quad \textit{Thrust}$$

$$F_2 = \rho\ [C_2 (V + \upsilon) \theta_c + C_3 (V + \upsilon) - C_4 (V + \upsilon)^2 + C_5 \quad \textit{Torque}$$

$$\text{where } \upsilon = \text{sqrt}\ [F_1 / (2\ \rho\ \pi R^2)]$$

The values for the constants were obtained by comparison with the LHX high fidelity helicopter model. It is conceivable that constants obtained by comparison with other models, or with flight test data would provide the relationships necessary to model different helicopters.

The thrust and torque relationships are especially interesting since they are in terms of clearly defined input values, such as pilot collective input, vertical velocity and atmospheric density. These equations were then used to calculate total body forces and moments, and thereby complete the aerodynamic reaction model.

Results

The generic helicopter completes all its processing in the alloted 33 milliseconds, compared to the 100 milliseconds required by the high fidelity helicopter model with the same control system. Off-line check cases** were.run to evaluate the time response of the two models to step inputs in all four control axes. The results of the aerodynamic angle comparison are shown as Figs.4 - 7.

As the check cases show, the response of the two models is very close, due in a large part to fine tuning the coefficients in the equations for the reaction model. Comparisons of positions, velocities, and altitude shows similar correlation. Comments from engineering pilots flying both models were very positive, and confirmed the close correlation of handling qualities that was inferred from the check cases.

In addition to achieving the computational and handling qualities goals, the generic model successfully integrates with the ARTI control system. All pilot relief modes were active, and responded as expected. Low frequency control oscillations due to the lack of resolution in the aerodynamics model were not apparent, although some small amplitude oscillations of a higher frequency were noticed in the check cases. These higher frequency oscillations were not noticed by the engineering pilot flying the comparison tests.

Application

Since the generic helicopter requires much less processing power than a high resolution helicopter model, it is an ideal candidate for a multiship combat simulation. The helicopter could be used either as a threat or friendly aircraft to enhance the tactical realism of the environment. Since the control system reduces the pilot workload considerably, a single pilot could fly the simulated vehicle, navigate, and operate the weapons system, making this model an ideal platform for testing of non-aerodynamic systems. If a helicopter pilot was not available then an engineer could easily learn to fly the helicopter model due to the user-friendliness of the ARTI pilot relief modes.

Finally, by changing the constants on the equations in the reaction model, the generic helicopter could be configured to represent specific helicopters, with some limitations. These helicopter models could also be used for training, workload evaluations, or tactical simulations.

* ARTI is the Advanced Rotorcraft Technology Integration control system which is the precursor to the MDHC Flight Path Command control laws. The original ARTI control laws were developed and tested by Benny B. Barnes, Gene Jackson, and Betty Neil.

** Data Collection and assistance was provided by Brett Hofstat.

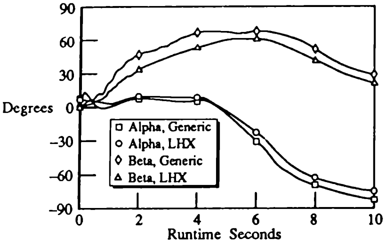


Figure 4. Aerodynamic Angles, Generic vs LHX
In Response to a Collective Input

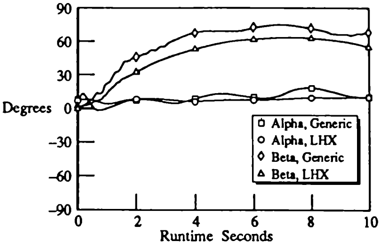


Figure 5. Aerodynamic Angles, Generic vs LHX
In Response to a Directional Input

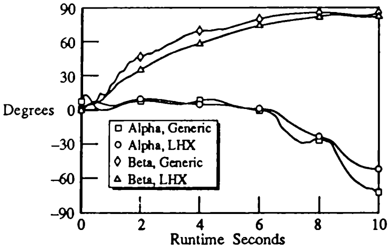


Figure 6. Aerodynamic Angles, Generic vs LHX
In Response to a Lateral Input

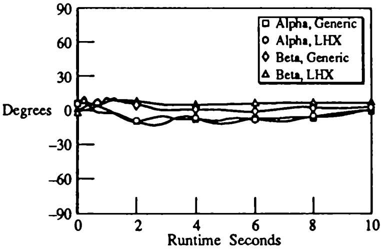


Figure 7. Aerodynamic Angles, Generic vs LHX
In Response to a Longitudinal Input

AIAA-92-4152-CP

A Mission Adaptive Combat Environment (MACE) For Fixed And Rotary-Wing Mission Simulation

David W Roden
David A Harrhy

Flight Systems Department
Defence Research Agency
Farnborough, Hampshire
ENGLAND

Summary

Advances in its design and performance have placed the modern graphics workstation in a truly competitive position in the simulation market-place due to its price-performance ratio and its ability to be cheaply networked. These abilities have contributed to the development of low-cost networked simulation. This paper describes one such networked simulation: MACE, a low-cost multiple man-in-the-loop simulation developed by the Defence Research Agency for its mission simulation requirements.

Introduction

For a number of years, simulation has been successfully used as a platform on which to assess and analyse various aspects of mission effectiveness. These aspects range from tactical deployment, through the man-machine interface, to individual aircraft module performance. However, modem avionics fits for combat aircraft require vast quantities of data regarding their combat environment. The avionics components, themselves, require a further mutual data interchange so that they may provide a degree of data-fusion. In future generations of combat aircraft, avionic systems are likely to become increasingly complex and interdependent as operational demands become more exacting.

These quantities of data flow have complicated the mission simulation task. Complex combat environments, containing air and ground based targets and threats, further complicate the task of simulating modern engagements. Adding the fact that multiple man-in-the-loop simulations provide the highest fidelity in combat mission modelling, it is evident that contemporary system implementations are likely to provide an expensive means of mission simulation.

However, the advent of the low-cost graphics workstation has provided a tool with which to reduce the cost of mission simulation yet keep a high degree of model fidelity through distributed model processing. The ability to network multiple low-cost workstations provides a powerful system with which to generate simulations, and this has led to the development at DRA of the low-cost, flexible mission simulation facility known as the Mission Adaptive Combat Environment, or "MACE".

MACE was originally developed as a fixed-wing air-air combat simulation facility, but more recently has been expanded so that it may encompass fixed-wing ground-attack and also rotary-wing simulation for anti-armour and air-air combat operations.

For a networked system to provide the flexibility and configurability necessary for a mission simulation facility, the following features are required:-

Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

- a distributed simulation to exploit workstation hardware currently available
- high-speed networking
- multi-player facility
- modularity
- capability to provide analysis facilities from which to generate results
- Reconfigurability to conform to any airframe/avionics fit
- Adaptability to any tactical situation

System Overview

The way that MACE has been designed encompasses all of these fundamental requirements. A description of MACE is given below. The description broadly addresses the fundamental concepts involved in the system.

Hardware Configuration

The backbone of the MACE system is a network of UNIX™ based graphics workstations and power server machines. The network common to these workstations is the ethernet® 10 Mbits/s Local Area Network (LAN). This network provides an excellent opportunity to modularise the simulation system in that a common data transfer protocol may be adopted by all entities involved in the simulation (see below).

For current requirements of MACE, the ethernet® is considered to have a sufficiently high bandwidth.

Model Representation

A prime requirement of the MACE environment as far as modelling is concerned is that it would be able to accept a variety of models written in a variety of languages. The reason for this is that existing models need to be available to the simulation facility in order to reduce development timescales. What is required is a method by which to modularise entities within the simulation and yet continue to keep a full data interchange between those modules.

Entity Modules

The way in which MACE sets out to modularise the entities is to use a common data packet communications protocol for all data interchange between entities within the simulation. The characteristic of the entity is determined by a separate software model. A common modular harness has been developed in which any entity type and the necessary communications could be encapsulated.

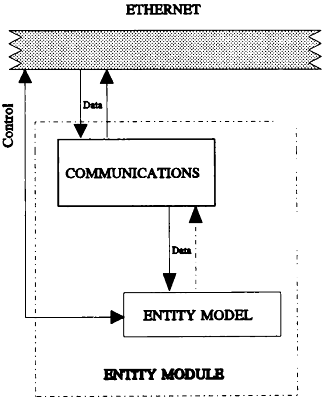


Figure 1 - Entity Module

The harness is controlled via an external control service, again via the ethemet® service to facilitate start, stop, freeze and reset functions from a central control station.

Communication Service

The MACE system provides the communication service through a data packet protocol called DDP. The DDP data packet is defined within the harness and contains a full array of information regarding that particular entity.

DDP packets can contain data referring to aircraft state vectors, projectiles and emissions, some of which data is directly used by the simulation and other data which is used for the analysis.

The communication hamess receives data from the owner entity and transmits it to all other entities in the simulation. Data received through the network is assimilated into a database storing the currently available data of each entity within the simulation. Incoming data is either added to the simulation database if it is a new entity or used to update stale data of an existing entity.

The database gives each entity access to any data pertaining to the simulation that it may require, through its own version of the database. The items of this available data that an entity may need to use are entirely entity-dependent, and this is discussed later.

Entity Modelling

Different entities require different consideration in the MACE environment. We will now discuss the way in which different entities use the DDP data and interact with the simulation. Let us consider some of the major types of entity we may encounter within a particular simulation:-

Players: Players may be of many types. In mission simulation players will generally be aircraft of either fixed or rotary wing. MACE is not limited to this, and makes possible the incorporation of ground forces such as tanks and SAM-sites.

Sensors: Sensors cover both active and passive sensors. These may include Radar, Infra-Red Search and Track (IRST), Radar Warning Receivers (RWR) and Missile Approach Warning (MAW).

Weapons: Weapons may include Guns and missiles such as Air-Air Missiles, Anti-Tank Guided Weapons (ATGWs) and Rockets.

C'measures: Chaff, flares, smoke and jammers may be represented.

Displays: Outside world visuals.

Intelligent Combatants: Stand-alone computer generated player models which include platform, sensor, weapon, and countermeasure modelling.

MACE Entity Construction

In this section we will examine the construction of the MACE modules, describing their functionality and their modes of interaction.

Players

MACE players use a generic flight model (different models for fixed and rotary wing, naturally!). The discriminating factor between MACE player handling lies in the Data Set that they use to determine the aircraft characteristics. These Data Sets contain data concerned with such issues as aerodynamics, fuel loading effects, engine characteristics and other aspects such as Radar Cross Section (RCS).

Different instances of the same flight model may be executed at different nodes of the network (or at the same node if required) and may be initialised with different data sets. This means that highly representative players of friendly and opposing sides may be generated within MACE with great ease.

The players, or rather the flight models, require input data pertaining only to flight control positions (such as throttle, stick, rudder, collective and cyclic) and output positional, attitudinal, and state data.

Figure 2 below gives a diagrammatic representation a player model.

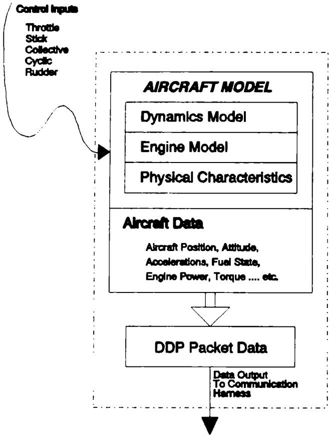


Figure 2 - Player Representation

Sensors

Sensor modelling can be very complex to perform. It is, however, a type of modelling to which MACE is ideally suited. Most sensor environments are complex, receiving data from a wide array of sources. A networked simulation is able to produce these sources in a truly representative manner.

Currently MACE uses a generic Radar model in much the same way as the flight models are used, ie. data sets provide the operational specification. The Radar model is therefore able to simulate a vast array of operational Radars, provided the data sets are available.

Using the example of the Radar, MACE sensor operation can be demonstrated. A Radar is an active sensor, ie. it stimulates the detectable objects through the emission of many Electro-magnetic pulses and measures the returned energy to determine the presence of, position, bearing and velocity of those objects.

The detectability of those objects is modelled through the knowledge of the position and velocities etc. of all entities in the simulation, and is subject to various influences. These will be RCS and may be countermeasures (chaff), close vicinity counterparts' emissions and reflections, and Radar Jamming. More importantly, the Radar functionality will be determined by the environment, ie. the 3-D space in which it operates. For example, a ground based simulation will require a ground-clutter model in its radar modelling.

The Radar is also producing an emission that can itself be detected by other sensors and identified - eg. by a Radar Warning Receiver. The DDP data packets facilitate the data interchange as discussed earlier. The sensor models themselves determine the extent to which they are affected by these emissions.

Weapons

Like aircraft, weapons are governed by a software model. Missiles, for instance, can use a model which may be generic or specific to type and which contains a seeker, motor, and dynamic fly-out model. A gun model is obviously governed by only a ballistic projectile/aerodynamic fly-out model.

However, differences between a weapon and a player are many. For example the weapon model is able to decide whether or not its deployment was successful, and can remove a player from the simulation (or temporarily disable them if they are not an airborne player). Also, if it is active homing or hot, the weapon model may provide emission data such that it may be detected by other sensors, for example a Missile Approach Warning (MAW) system. Finally, the weapon model, if employing a seeker, must take into consideration, the environment which surrounds it. Decoys, jammers, other players and the clutter environment will all have an effect on its operation.

Countermeasures

Countermeasures for the main part are treated like a projectile. A simplified dynamics

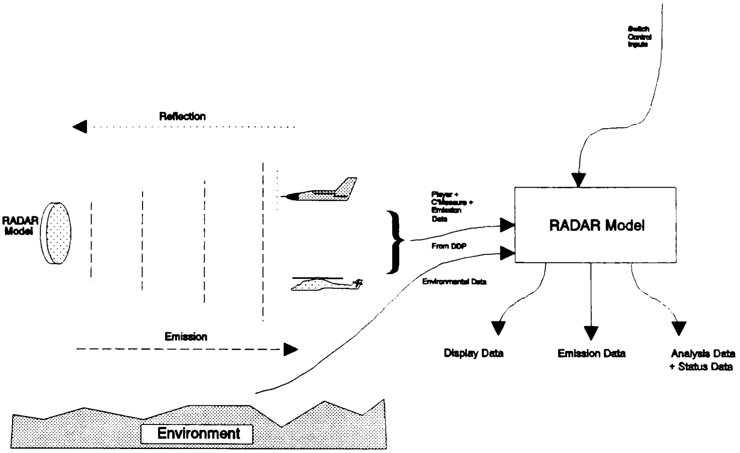


Figure 3 - RADAR Module

model governs the dispersion and position of chaff and smoke, and the position of flares. The presence of jammers and decoys can also be modelled. The sensor and weapon models have the responsibility to act representatively in their presence.

Displays

The most important aspect of the simulation as far as manned interaction is concerned, is the display suite which is available to a pilot. It is impossible for the pilot to interact with the simulation unless there is an information feedback to him.

In a simulation there are many display surfaces that may be required. These are generally flight instruments, sensor displays, and an outside world display.

As the DDP database contains constantly updated data pertaining to attitude, position, status, and types for all players, weapons, and decoys in the simulation, it is relatively straight-forward for a display generator program to transform this data to a display format for one or more of the display surfaces. For instance, a baro-altimeter display simply requires access to the baro-alt variable in the DDP database before it has all the data needed to produce its display. A more complicated display such as an outside world display requires full positional, attitudinal and type data for all of the visible entities within that particular simulation. Additionally it requires a copy of the environment database describing the physical environment in which the player exists.

Intelligent Combatants

So far in this paper only manned interaction with the simulation has been discussed. Manned interaction has the advantage of realistic scenario input, but two main drawbacks:-

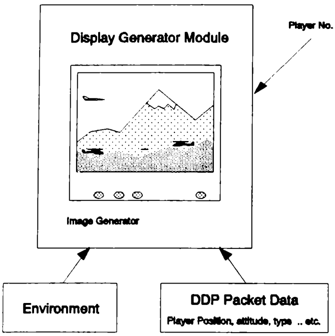


Figure 4 - Display Module

1) Additional hardware is required for each additional pilot station

2) More subject pilots are required per scenario

In some cases it is essential to have a manned station when that player is generating significant data, but players on the periphery of the combat can often be replaced with computer-generated IKBS players. More importantly, players of types for which manned interaction is not available, for example a tank, can be more economically generated using intelligent combatant models of them.

An intelligent combatant model can be integrated in a straightforward manner into the MACE environment by encapsulating it in a communication harness with DDP.

The advantage of the MACE networked simulation is that pilots at the manned stations can be made completely unaware that the players are artificial, thus increasing the realism in workload.

Interfaces To MACE

So far only the simulation modules have been discussed, along with the concept of the MACE networked simulation. Interaction with the simulation is of paramount importance, for both conveying data to the simulation and extracting data from it.

Pilot Console

For dealing with pilot input, realistic inceptors were constructed for the pilots to interact with. These inceptors are representative of real aircraft cockpit controls. Low-cost non-flight-worthy controls are used to provide the hand-controllers and these are mounted on a purpose built chassis which when in use is connected to a desk with a workstation on top. The controls used vary in their arrangement between fixed and rotary-wing for obvious reasons, though the same physical controls are used.

Commander / Supervisor Console

During simulation run-time it is often advantageous to have the facility to oversee the combat. This can be helpful for many reasons, but a common use is for an independent supervisor to oversee the simulation as it happens. It may also be employed by a squadron commander with a C^3I interface (eg. AWACS) to the simulation.

The supervisor console is a 3-D representation of the simulation. All visible entities are displayed interdependently by position, attitude and type relative to the outside world (generally a representation of the outside world is all that is required). The supervisor may query certain elements of data from this display for various players. The supervisor console uses DDP to obtain access to the simulation data. The console is, however, limited to viewing only.

Logging / Analysis Facilities

In any simulated combat interaction it is necessary to know afterwards what went on. Although some of this analysis may be performed at run-time, for the most part, it will be carried out at a later date. For this to occur a data logging mechanism must exist.

One advantage of MACE, as already mentioned, is that it allows any processor on the MACE network access to the simulation data via DDP. This simplifies the logging task drastically as all incoming data is fresh, and is therefore all that is required by the logging process. The only drawback is that with a large simulation, the data files logged for any combat are likely to be huge, so a large fast access medium is required for logging. Fortunately, most modern workstations are equipped with this storage capability as a standard configuration.

Once stored, analysis of data can be achieved in various ways. For tactics evaluations a replay facility is often the most appropriate method. A version of the supervisor console reading logged data rather than the ethernet® is available as an effective replay facility. Additionally graphical interpretations of data can prove useful in such areas as vehicle and sensor performance. Data can also be condensed down and converted to a spreadsheet format for copying across to Personal Computers.

Concluding Remarks

This paper has described the MACE simulation system developed by the Defence Research Agency, Farnborough, England. The system is a distributed simulation capable of producing a manned multi-player engagement for both fixed-wing and rotary-wing aircraft. The hardware consists of an array workstations operating over an ethernet® and providing a powerful graphics capability. These support a reconfigurable and low-cost simulation environment.

Use of this system has proved it to be a valuable platform for mission simulation, analysis and tactical evaluation.

The nature of MACE renders it applicable to all warfare environments with the only requirement being the generation of suitable entity models.

AIAA-92-4154-CP

A DIGITAL THREAT GENERATION NODE FOR A SIMNET/DIS TACTICAL AIRCRAFT NETWORK

Frederick B. Fleury
Douglas S. Goold
General Dynamics Corporation
Fort Worth Division
P.O. Box 748, Mail Zone 5944
Fort Worth, TX 76101

ABSTRACT

The Flight Simulation Laboratory (FSL) at General Dynamics, Fort Worth Division, is continuing to develop a SIMNET/Distributed Interactive Simulation local area network. This network was developed to support engineering studies and aircrew training simulation. The latest upgrade to this network includes a digital threat generation node to provide a unique combination of evaluation and tactics training scenarios. Although the SIMNET/DIS protocol provides an inexpensive approach to networking many simulators together, there is a need for an automated force structure in addition to the man-in-the-loop network nodes. This capability will allow increased flexibility in the blue vs red test and training scenario combinations, while reducing required test support resources.

The threat generation model used for this study was re-hosted from a Harris H-1000 computational platform to a Motorola 88100 VME computer system. This threat model has been tested and used extensively in the FSL for engineering studies and F-16 aircrew tactics training. This paper will discuss the initial configuration developed at General Dynamics, which used a dual simulator F-16 configuration to test the SIMNET protocols for tactical aircraft. The results from this baseline developed the rationale for additional investigation into the use of a threat node on the network to generate Airborne Interceptor (AI) aircraft and Surface-to-Air missile sites.

The three steps used to integrate this threat node on a network will also be discussed. These steps are: 1) the threat model re-host task between the two computer platforms, their differences and performances, 2) the integration techniques used in installing the digital threat generation node on the General Dynamics local area simulation network, and 3) the addition of any protocol extensions or computational platform modifications necessary to complete the integration.

INTRODUCTION

General Dynamics' interest in the SIMNET/DIS approach to network simulation began with discussions with the Armstrong Laboratory (ALHRA) personnel at Williams AFB.
These discussions concerned the need to test many of these protocols for tactical aircraft. Dr. H. Bell and Maj. Kamrowski offered to loan General Dynamics network interface equipment to investigate the SIMNET protocols in the FSL. There was some initial skepticism by the FSL of the viability of using an ethernet network to provide real-time updates to high fidelity flight simulations. These concerns centered around the following: update rate, bandwidth saturation, position errors induced from the extrapolation routines, and visual fidelity of the networked aircraft.

However, there were many advantages to researching these concepts and collecting hard data specifically for tactical aircraft simulation.

This was also seen as an opportunity to find a low cost solution to network F-16 based training system products. Similar communication strategies could potentially be used as a standard means to communicate between the various engineering simulators in the FSL. If validated, these methods could further be extended to include the other Fort Worth facility laboratories.

The investigation was broken down into three concerns for using the SIMNET/DIS methods. First, could the SIMNET Network Interface Units (NIU) connected through ethernet provide updates to the host computer systems at a sufficient rate? Second, would the visual scene presentation of the networked aircraft be of high enough fidelity for gun sight evaluations? Last, could actual end-game calculations be performed to determine the outcome of an aircraft engagement?

1 v 1 SIMULATION CONFIGURATION

Figure 1 shows the 1991 baseline configuration that was developed in the FSL. It includes two F-16 high fidelity flight simulators using Harris H-series host computer systems, Silicon Graphics 4D-series graphics generators for cockpit displays, and an Evans and Sutherland CT-6 visual scene generation system.

The simulation host computer systems were connected using ethernet to the Network Interface Units (NIU) Motorola 147 processor. This connection was updated at the host computer's maximum frame rate of 50hz. The NIUs used the VME backplane as the means of communicating the SIMNET protocols. The ethernet packet communications protocol was maintained on the VME backplane to allow the configuration to use a Local Area Network (LAN) between the NIUs in the future.

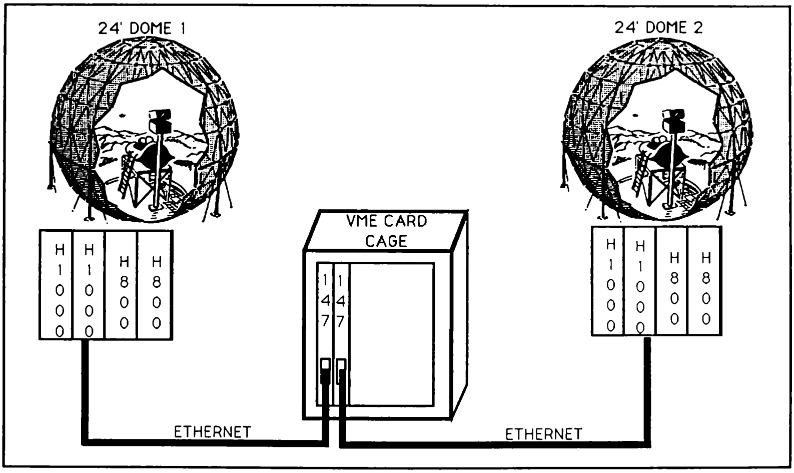


Figure 1. Initial Dome vs Dome System

The NIU Motorola MVME-147 processors used a Software Components Group (SCG) pSOS+ real-time kernel as its operating system. The NIU source code was provided by ALHRA and modified for use with pSOS+. A SIMNET conversion routine was written to run on the Harris host computer system to map the F-16 simulations ownship variables into the SIMNET protocol structure. This software was integrated on both F-16 computer systems to provide the 1 v 1 test configuration.

The following extensions were added to the SIMNET protocol:

1) linear accelerations - these were added to allow for future dead reckoning research using higher order algorithms or for use by visual systems.

2) orientation velocities - for future dead reckoning research.

3) modified FIRE PDU - to avoid saturating the network while firing high rate of fire weapons like the F-16 20mm cannon.

The baseline testing involved the visual tracking of the networked wingman aircraft to assess visual fidelity. The lead aircraft was followed closely through a series of typical maneuvers. Data was captured at 50Hz for each F-16 simulator's position, rotation, velocity and acceleration components for later quantitative analysis. A NIU threshold of 1 meter was chosen for the positional axis components and 3 degrees for the rotational axis components.

Figure 2 plots one of the F-16 ownships X and Y position as the solid line and its resulting target image X,Y positions plotted as the dotted line as see from the networked F-16's viewpoint. Analysis of this plot indicated a maximum delta position of 232 feet from the source F-16 simulation data.

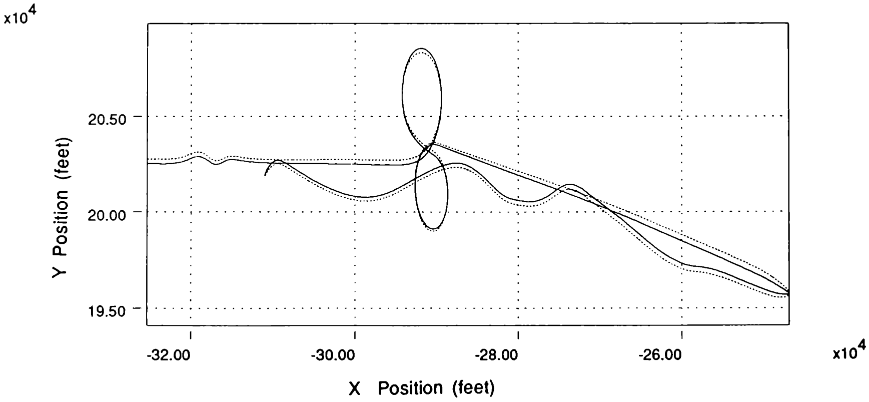


Figure 2. X vs Y position for Source Aircraft and Target Image

Further breakdown of this data plotted in Figure 3 indicates that the X positions for the source and the resulting target image plotted verses time overlay one another with the threshold allowance of 1 meter.

Plotting the Y positional data shown in Figure 4 indicates a large delta between the source F-16 positional data and the data used to generate its image to the networked entity. This was tracked down to problems in the NIU code associated with deriving the Y velocity terms for use in the dead reckoning algorithms. Investigation into the NIU code will correct this problem with the Y velocity derivation and bring the Y positional data within the desired 1 meter threshold. Roll reversal problems were also encountered but were tracked down to a error in the arc tangent function.

Further testing of the Network Interface Units found that each local entity required 1.2 msec, and each remote entity 1.1 msec of the 20 msec frame time to buffer in the host data and perform the dead reckoning functions. As shown in Figure 5 this processing was found to be linear for each entity added to the network. The plot also shows that the highest number of entities processed by the NIU is approximately 10-12, allowing the send and recieve software and the pSOS kernel their 6 msec overhead time.

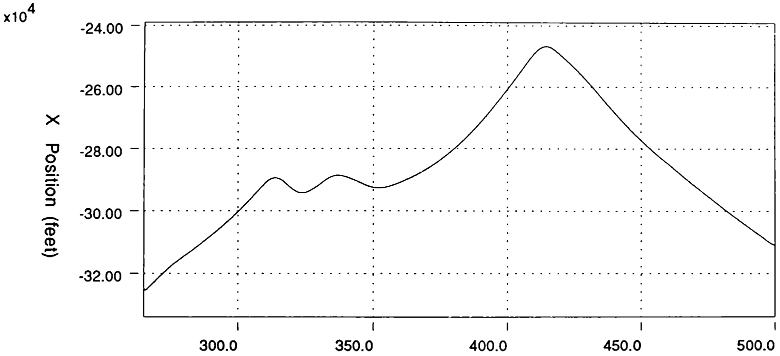


Figure 3. X position vs time for Source Aircraft and Target Image

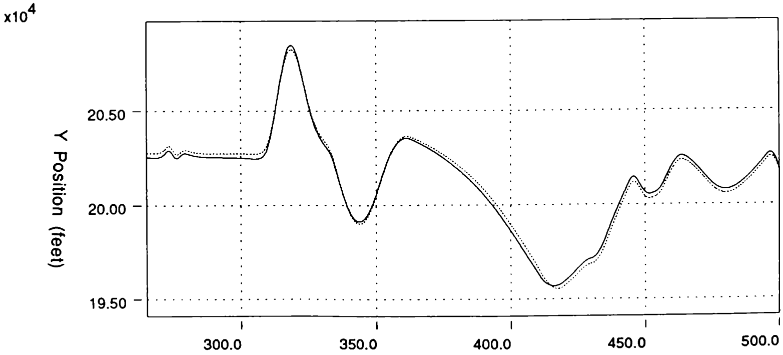
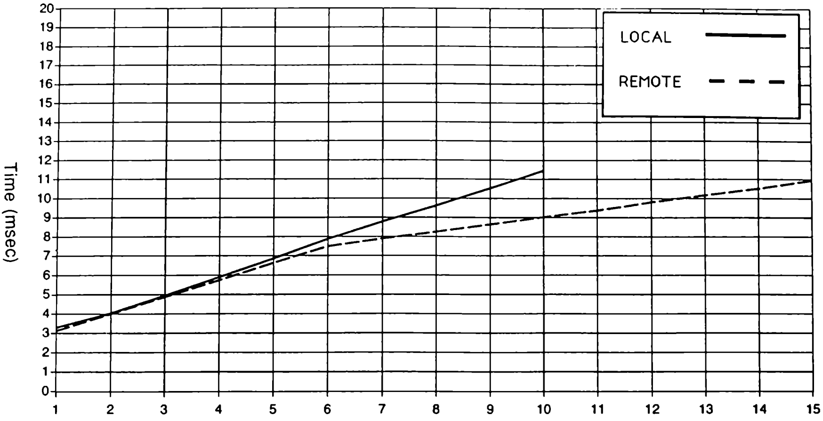


Figure 4. Y position vs time for Source Aircraft and Target image



Number of Host Entities

Figure 5. MVME-147 NIU Timings for Host Entities
(1m and 3 deg. Thresholds)

Figure 5 also indicates that the number of remote entities has been limited to 6 vehicles in the NIU code. For each entity after the limit of 6, requires 0.4 msec just to reject the entity.

THREAT NODE DESIGN

Based on the 1991 tests results, a decision was made to investigate the possible problem of NIU saturation along with end-game capabilities with our 1992 efforts. The test configuration would require the use of the threat model developed in the FSL. This model generates digital target aircraft and surface-to-air missile sites. The model has been used extensively for engineering evaluations of weapons algorithms and survivability studies within the FSL for many years. We decided to rehost this threat model to a separate computer platform from our Harris host system. We would then add this new processor to the SIMNET network as a third node.

The host computer for this node was the next consideration. The laboratory has a variety of platforms available including: Harris H-series, Encore 32 series, SUN Sparc 1+, and Silicon Graphics 4D series computers. The ALHRA personnel also offered to provide a Motorola MVME-188 Quad processor board for this development. This MVME-188 processor was seen as an excellent opportunity to research VME host computer systems to determine their performance and limitations. The Silicon Graphics systems was reserved as a second platform, if the Motorola MVME-188 did not pan out. A SCG Passport target software package was available in the FSL for the 88100 processor. A SUN version of the Green Hills FORTRAN cross-compiler was also available in the laboratory. These three components completed the tool kit necessary to develop and run real-time tasks on the Motorola MVME-188 processor.

IMPLEMENTATION

The first steps in rehosting the threat code was to study the differences between the FORTRAN implementation on the Harris computers systems and the Green Hills compiler. These differences were found to be very close to the list developed for re-hosting code from the Harris to the Encore 32 series systems. The differences included 24-bit to 32-bit word conversions and the math library calls. Software was developed to make these code changes automatically as the files were passed through this filter software. The filter also changed all of the occurrences of the DATAPOOL structure used on the Harris systems to use INCLUDE files. A shared memory area was then set-up in the 188's local memory to provide a shared common area for all tasks to use. This was necessary since the Harris version of the threat model runs the target aircraft and the SAM sites as separate tasks. It was our desire to maintain this same approach. For the next step, a root task was written in 'C' that would call the modules and provide the required 50Hz frame timer on the 188. The ability to measure the execution speed of these processes was also implemented in this root task code. A method of validating the performance of the threat model against the original Harris model was then needed. This was performed by two methods, quantitative data analysis of the threat model variables memory contents and subjective evaluations from a graphical output.

It was decided that a SG 4D/25 running our Mission Overview (MOV) software would be useful for the second method. This MOV model graphically shows position and other data associated with the F-16 aircraft, the air targets and the SAMs. For quick integration, the SGI 4D/25 was connected through a RS-232 port to the MVME-188. This test configuration is shown in Figure 6. Figure 6 also indicates that the SIMNET network portion of this configuration is all performed on the VME backplane. It was our original intent to use the Motorola 147 ethernet ports as the SIMNET connection and CMC ethernet cards would then be used as the host connection. In implementing this approach we found severe speed limitations with the MVME-188s ability to write to VME memory. We were finding 1 usec per tranfer to be the typical time for the MVME-147 processor and the Harris Night Hawk using a 88100.

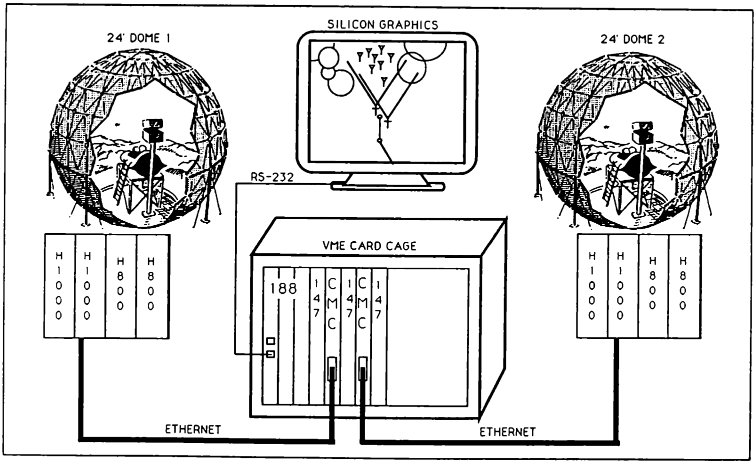


Figure 6. Dual Dome System with Threat Node

However, for the MVME188, the transfers were taking between 6 and 8 usec. Using a 1500 byte ethernet packet, it would require more than half of the 20 msec frame time just to write to the CMC ethernet cards memory. We found this usage of the frame time to be unacceptable. Wishing to move on with our research, it was decided to let the MVME-147 processor grab the memory contents from the MVME-188s memory and then provide it for NIU processing. We are still researching the MVME-188 write time limitation with the Motorola technical personnel.

One more configuration is also possible for using the traditional ethernet SIMNET configuration. This would require the CMC ethernet cards 68020 processor to get the data out of the MVME-188s memory and then provide it to the ethernet chip for transmission. In the interest of providing research data, the current configuration will be used for the first round of testing this year. Modifications to the system will include these other configurations, as soon as the solution to the MVME-188 transfer problem are found.

We are still currently using the SIMNET version 6.6 with our GD extensions to provide the transmission protocol between the NIU nodes. The DIS ver. 1.0 protocols will be implemented this summer for testing and interface evaluations. A comparison of the two protocols for this threat model interface will be presented at the conference. The test data for this years initial configuration and any change in findings on the Motorola 188 will also be presented.

CONCLUSIONS

Our initial testing last year found the SIMNET approach to be acceptable for aircraft concept evaluations and training. In general, the following observations were made.

1. The visual fidelity of the networked aircraft was sufficient for target tracking and wingman formation flying. Closer threshold settings would be required for true end-game and survivability analysis work.

2. The current MVME-147 processor based NIU configuration may not be robust enough for multiple target or threat site engagements found in tactical aircraft scenarios. A Motorola MVME-167 processor will be investigated this year as a possible solution. The incorporation of the NIU code on the MVME-188 processor is also being considered.

3. The use of a MVME-188 card for the threat module is a very enticing solution from a systems engineering standpoint. The advantages include the low cost of a very fast processor in a VME card cage and flexible system configurations. Drawbacks include a limited selection of software real-time kernel support for this product, immature compiler software, and the possible write to memory speed problems encountered during this implementation.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Herb Bell, Maj. Mike Kamrowski and Capt. Rick Rodgers from ALHRA - Williams AFB for there assistance and the loaned hardware. We would also like to thank David Parker and Travis Teselle for their endless efforts in integrating the MVME-188 and CMC ethernet processors. Steve Hawkes for his never ending knowlege of compilers and other magical software. Many thanks to Don Smart, David McKensie and Bill Abshire for their guidance and editoral comments.

AIAA-92-4155-CP

A PROTOCOL CONVERTER

FOR

NETWORKED AIR DEFENSE APPLICATIONS

Daniel A. Bradford • Danielle M. Eriksen • Alan M. Thibodeau
Lockheed Sanders, Inc.
Nashua, New Hampshire

Huat K. Ng
Institute for Simulation and Training
Orlando, Florida

ABSTRACT

The standardization of training/simulation protocols through the SIMNET/DIS concept creates a unique opportunity to maximize the productivity of existing fielded systems. The adaptation of trainer protocols to SIMNET/DIS protocols will allow trainers to be integrated with distributed simulation networks, thus expanding SIMNET/DIS to better simulate and model the complete air-land battlefield. The focus of our work has been in the design and development of a general form protocol converter to adapt air defense trainers/simulators to a distributed simulation network. The air defense protocol converter incorporates generic air defense processing models, a menu driven user interface for customizing the protocol converter, and an object oriented design with clearly defined interfaces between models. The protocol converter houses all of the SIMNET processing needed for air defense systems to participate in a SIMNET exercise. In addition, we have defined a minimal set of messages between the air defense trainer and the protocol converter, which contains typical information available to the trainer. The protocol converter uses information from the generic interface message set and the user interface files to provide an air defense application with an intelligent interface to SIMNET, which can be easily expanded to address DIS.

INTRODUCTION

Advances in simulation technology coupled with a decreasing defense budget have created an environment which warrants increased simulation modeling for battlefield environments and an increased desire to upgrade existing fielded systems. Work initiated under the DARPA SIMNET (SIMulation NETwork) program has demonstrated the ability to develop simulations, force on force, that can create a realistic electronic battlefield environment. The Battlefield Distributed Simulation - Developmental (BDS-D) program of STRICOM has begun to address extensions that will create the entire "electronic battlefield." With the definition of the Distributed Interactive Simulation (DIS) protocols reaching a version 1.0 milestone, the remaining tasks will disseminate the architecture to platforms, systems and technologies that will support the entire electronic battlefield.

The use of SIMNET-based protocols has resulted in a significant new approach to training both large and small combat units. SIMNET is currently operational at training sites in the United States and Europe. Simulators are linked by a common real-time data communications network. Crews of these simulators can see and interact with each other against accurately scaled opponents on the same realistic battlefield to provide effective force-on-force training in a combined arms battlefield environment. The standardization of training or simulation protocols based on the SIMNET and DIS concept creates an opportunity to maximize the effectiveness of fielded training systems to provide large-scale training exercises without incurring the costs of transporting large numbers of personnel and equipment. Modification of existing fielded training systems to implement appropriate SIMNET protocols is a cost-effective way to upgrade existing assets to provide new functional capabilities while retaining currently implemented training functions such as instructor monitoring, student assessment and grading.

The adaptation of trainer-specific protocols to SIMNET protocols will allow existing trainers to be integrated with SIMNET networks, thus expanding SIMNET to better simulate and model the complete battlefield. Additionally, the quality of the local training system is increased as SIMNET engagements provide a more realistic range of command and control as well as combat service support elements found in actual military operations.

While many existing training and simulation systems are based on incompatible hardware and software architectures, suitably capable trainers can now be adapted to interact in the SIMNET environment through the use of a protocol conversion process. This process can be as basic as simply translating relevant trainer-specific information to and from SIMNET protocols, or can be made more intelligent to house additional SIMNET/DIS processing which the trainer/simulator is not concerned with.

BACKGROUND

The SIMNET/PATRIOT Protocol Converter (SPPC) project, funded by STRICOM, has researched the feasibility of linking a fielded air defense trainer/simulator to a distributed simulation network through the use of an intelligent protocol converter. The research, design and implementation of the protocol converter are the focus of the program. The air defense trainer (ADT) used in this research is the PATRIOT Operator Tactics Trainer (OTT) originally developed by Lockheed Sanders, Inc. for the US Army. The trainer has gone through recent modifications to incorporate technology upgrades and is presently fielded in Japan. As part of this upgrade, a "virtual console" was created which is a workstation based hardware simulation of the actual console. It is this virtual console and the Japanese version of software which have been used as the original baseline system in this research. The simulation networks researched are SIMNET (SIMulation NETwork) as described in "The SIMNET Network and Protocols Report No. 7627", and DIS (Distributed Interactive Simulation) as described in the document entitled "Military Standard (Final Draft) Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation, publication IST-PD-91-1".

The current Lockheed Sanders PATRIOT OTT Fire Platoon (FP) was examined to determine whether it was feasible to connect it to SIMNET. The PATRIOT OTT FP software did not have the capabilities required to communicate directly to SIMNET or a Protocol Converter. It was configured to communicate with an instructor's console, and to simulate air defense using preprocessed target data. Under internal Lockheed Sanders funds, modifications were made to allow communications with the protocol converter and to allow for the real time operation of the FP software.

Detailed analysis of the SIMNET PDUs identified which fields in each message could be supplied directly by air defense trainers, and which fields from incoming messages would be needed by air defense trainers. Because air defense trainers will potentially vary widely in their level of fidelity and their capabilities, we attempted to keep the number of parameters to a minimum. This has maximized the intelligence and processing required by the Protocol Converter itself.

The SIMNET Protocols have been designed mainly for visual simulations; therefore, the message sets do not have complete data that would normally be required for sensor-based detection of airborne threats. The Protocol Converter development needed to compensate for this lack of information, and calculate default data for both incoming and outgoing PDUs.

After identifying those fields needed by the Protocol Converter to correctly process SIMNET PDUs, a generic interface was designed for communication from the Protocol Converter to air defense trainers. Use of the Protocol Converter to participate on SIMNET now requires that a trainer is modified only to implement this interface, which is far less complicated than the complete set of SIMNET protocols.

APPROACH

Our design has addressed two primary problems related to distributed simulation. First, how to modify existing, fielded training systems to be SIMNET compatible. Second, how to overcome the wide area network (WAN) implementation issues, such as bandwidth limitations and network delay. The design has been driven by our desire to minimize the changes to the existing trainer software and hardware, isolate the trainer local area network (LAN) traffic from the SIMNET LAN, and to develop modular/generic software.

To minimize the changes to the existing software/hardware, our first approach was to listen to the trainer network for SIMNET applicable information, read it off the network, and build PDUs to be sent to SIMNET. This approach made use of existing trainer messages where possible, with modifications to existing messages where necessary. New messages would only be created when the information was not on the trainer network at the required time.

This approach dictated that we have a working knowledge of the internal trainer software operation to allow us to create the appropriate SIMNET PDUs. It also assumed that the majority of the required information was present on the network at one time or another. If the information was not available, considerable changes might have had to have been made to the trainer, which violated a key design goal. If a single event on the trainer LAN did not correspond to the issuance of a SIMNET PDU, history might have had to have been kept at the Protocol Converter (PC) to determine when a PDU should be sent. This approach was not pursued due to the constraints that it placed on both the trainer manufacturer and the protocol converter.

The present design is based on the protocol converter being placed on the SIMNET LAN, with a low speed WAN connection' between it and the air defense trainer (see Figure 1). The protocol converter contains several models that perform air defense related processing. The design goal of minimizing changes to existing software has been met by defining a generic interface between the ADT and the PC. This allows the trainer manufacturer to implement the interface in the most efficient, cost effective manner possible. It also eliminates the requirement of having an understanding of the trainer software, and of the trainer having SIMNET relevant information on its network at specified times.

The design of the protocol converter also addresses the problem of WAN bandwidth and network delay for distributed simulation. By locating the protocol converter on the SIMNET LAN and connecting it to the trainer via a low speed WAN, we have been able to reduce the network bandwidth requirement and associated delays. This is possible because the time-critical processing related to air defense operations has been moved from the trainer location to the SIMNET location. These processing models have been designed to be generic and customizable to allow the maximum flexibility for use with air defense trainers/simulators. The co-location of the PC to SIMNET and its gateway design provide separation of the trainer LAN traffic from that of SIMNET, thus reducing the potential added traffic on the SIMNET LAN.

PROTOCOL CONVERTER DESCRIPTION

The Protocol Converter consists of six major software components; a Man-Machine Interface (MMI) - which is a separate executable from the actual Protocol Converter, a Protocol Translator, a SIMNET Vehicle Model, an ADT Vehicle Model, a Detection Model, and a Missile Model. All of the models have been implemented in Ada on a SPARCStation running Unix. The Ethernet communications portion of the

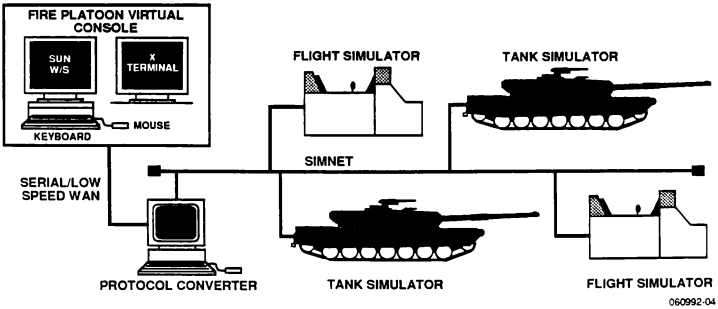
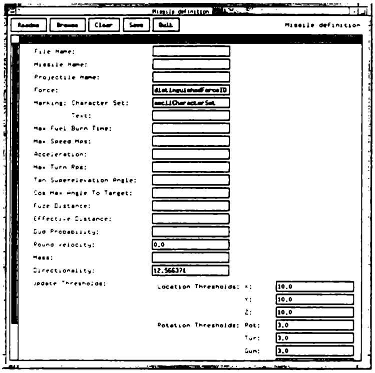


Figure 1. System Configuration

PC was implemented in 'C'. The MMI was also implemented in 'C', using the X Window System for the actual user interface.

To accommodate different air defense trainers, and also to allow individual air defense trainers to test different tactical strategies, there are several portions of the Protocol Converter which are user-configurable upon program start-up. The parameters are configurable through a menu driven interface which we have called the MMI, and are entered to the Protocol Converter in a format consistent with SIMNET PDUs whenever possible. For example, a vehicle's default appearance will be described in terms of Guises, Appearance, Markings and Capabilities. Parameters which describe how the detection model and missile model will function, such as subvolume ranges and missile fuse distance, are also configurable, even though they might not be used for PDU generation. A sample menu from the MMI is shown in Figure 2. These configuration parameters may still not allow all air defense trainers to use this protocol converter. They should, however, allow the flexibility needed for trainers and simulators of radar-based systems to easily use the system.



File Name:
Missile Name:
Projectile Name:
Force: distinguishedForceID
Marking: Character Set: asciiCharacterSet
Text:
Max Fuel Burn Time:
Max Speed Mps:
Acceleration:
Max Turn Rps:
Tan Superelevation Angle:
Cos Max Angle To Target:
Fuze Distance:
Effective Distance:
Dud Probability:
Round Velocity: 0.0
Mass:
Directionality: 12.566371
Update Thresholds:
Location Thresholds: X: 10.0
Y: 10.0
Z: 10.0
Rotation Thresholds: Rot: 3.0
Tur: 3.0
Gun: 3.0

Figure 2. MMI Sample Menu

The first component of the Protocol Converter itself is the Protocol Translator, which provides a bi-directional translation between SIMNET format and the Protocol Converter's internal format. All messages received from SIMNET are examined and either sent to the appropriate function of the Protocol Converter or discarded. Messages from the Protocol Converter are translated into the proper SIMNET format and padded with the appropriate headers and any additional default data from the MMI that the ADT could not supply. A change to a different protocol, such as DIS, will have the most effect on this component.

The remaining components of the PC are divided according to the objects that they represent in the simulation world, that is, ADT Vehicles, SIMNET Vehicles, missiles, and radars (or other devices which perform detection). The ADT Vehicle Model and the SIMNET Vehicle model are mainly responsible for maintaining data regarding ADT and SIMNET vehicles and processing them according to the SIMNET protocols. The missile model and the detection model use information from the ADT and SIMNET Vehicle Models for their respective processing (see Figure 3).

The SIMNET Vehicle Model maintains the current information for all of the SIMNET vehicles of concern to the ADT. These are normally only airborne threats, but may also be land vehicles. It receives Vehicle Appearance PDUs from SIMNET, and first performs some basic Field of View filtering to determine whether the vehicles should be maintained or not. Potentially detectable vehicles are maintained in a list according to SIMNET protocol requirements - that is, their positions are dead reckoned until the next update is received, and they are removed from the list if updates are not received within the required time frames. This list is traversable by the other models of the PC that need information about vehicles playing on SIMNET. If visual updates are required by the trainer, vehicle update messages will be sent from the SIMNET Vehicle Model. These vehicle update messages would also be used in the absence of the Detection Model (detection processing being performed at the trainer), or if the trainer used both visual and non-visual detection methods.

The ADT Vehicle Model maintains a list of vehicles that are part of the air defense trainer, such as radars and launchers. Vehicle Appearance PDUs are sent to SIMNET for each vehicle in the list according to the SIMNET rules for static vehicles. Damage assessment is performed upon receipt of Impact, Indirect Fire, and Collision PDUs. Control PDUs from SIMNET, such as Activate and Deactivate messages, are also processed and responded to. The ADT is informed if a vehicle is destroyed or deactivated by SIMNET.

The Detection Model is currently implemented as a generic radar model, although the package specifications were designed to hide this fact as much as possible, and the bodies could therefore be implemented to simulate another form of detection. The model is capable of handling multiple radars, each with multiple subvolumes for search and track processing. The detection model scans the vehicles maintained by the SIMNET vehicle model, performing a series of tests on the vehicles to determine whether each is detectable. If detectable, the vehicle becomes a track and an update is sent to the trainer. The Detection Model sends Radiate PDUs to SIMNET as required by the protocol.

The Missile Model computes the missile position as it chases the target, and sends update messages to SIMNET and the trainer on current positions. The Missile Model receives engagement messages from the trainer for an unlimited number of targets. A target can be either a track from the Detection Model or a target that was engaged visually and maintained by the SIMNET Vehicle Model. The model also handles outgoing SIMNET PDUs, including Fire, Vehicle Appearance, and Impact PDUs.

The level of fidelity for simulating the Missile Model and the Radar Model were major issues of concern for the SPPC project. The simulation had to be realistic enough to mimic the tactical system in the training environment, yet fast enough to perform on a single processor within the real-time constraints of a SIMNET exercise. These models will be discussed in greater detail below.

GENERIC RADAR MODEL

An investigation of radar modeling was undertaken to determine what level of fidelity was appropriate for the Detection Model of the SPPC, and whether any suitable software already exists. This investigation mainly targeted the Lockheed Sanders' Technical Library and Modeling Labs, and revealed that the majority of the radar models are engineering models that typically run in batch mode and generate a very high fidelity output. While the detailed modeling can theoretically provide small computational errors under common circumstances, it is felt that some of the computational corrections to the general radar equation are not needed for training. Documented algorithms of generic radar simulation and the equations provided in basic radar textbooks provided the fundamentals required for specifying a less accurate model that could be executed in real-time, even under heavy target loads.

Using the knowledge gained from this investigation, we designed a generic radar model which utilizes a multi stage process to determine the existence of tracks. One process ascertains if a target falls within

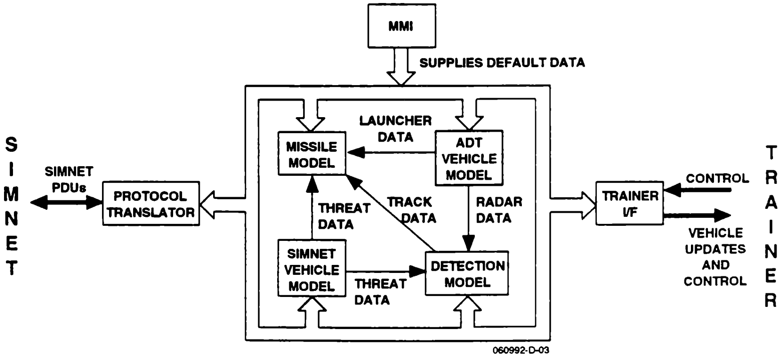


Figure 3. Protocol Converter Component Relationships

the appropriate search or track coverage area, and also which targets are illuminated. Another process determines the intervisibility between the ADT radars and targets. The final process determines if sufficient target signal exists for track identification, while accounting for some signal losses, as well as jamming and chaff effects. This processing should be done at the Radar Scan Rate for the particular radar.

A primary objective of the Generic Radar Model for the SPPC was to be able to handle an unspecified number of radars, as well as vehicles and missiles, in real time. Principally due to this reason, the algorithms of the Generic Radar Model do not account for many potential factors affecting the received radar signal, such as clutter. Environmental effects that have been left out are air temperature, pressure, humidity, the curvature of the actual radar beam due to refractive effects, and atmospheric losses due to beam scattering. Other effects ignored are radar specific and account for the physical characteristics of the transmitter and receiver antennas, and losses of the receiving equipment. Finally, the Generic Radar Model is based on a simplistic mono-pulse radar; it does not provide for a Moving Target Indicator discernible by a Doppler radar.

The Generic Radar Model is parameter driven in order to represent many types of air defense radars. These parameters are provided to the generic radar model using two different methods. The first method provides parameters through the use of the MMI, as mentioned above. These defaults may be superseded by the second method of parameter specification, via messages from the ADT. This method allows for the ADT, if it is capable, to dynamically create and modify simulated radars. This method should be utilized if possible, since a radar's parameters will normally change during the execution of a scenario.

Stages of Processing

The stages of detection processing are illustrated in Figure 4. The Basic Angular Coverage Check is the first test for each vehicle, and determines if it falls within the minimum and maximum azimuth and elevation of the combined search and track coverage areas. As search and track subvolumes (explained in a following section) are added, modified, or deleted, the overall minimum and maximum azimuth and elevation for these areas are maintained. This overall coverage area is used as a coarse filter to quickly eliminate vehicles that are neither detectable nor capable of directly jamming the radar. Vehicles that fall outside the overall coverage area are excluded from further processing.

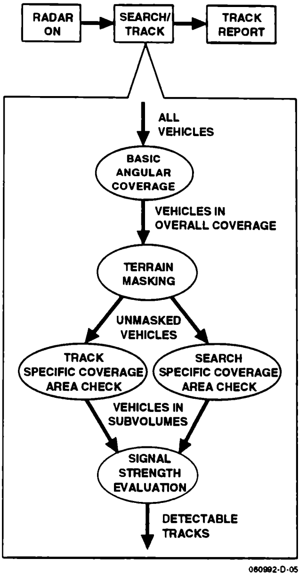


Figure 4. Radar Model Processing Diagram

Next, the Terrain Masking Check tests to determine if a vehicle's detection or jamming signal is blocked from reception by terrain. To perform this processing, a Terrain Masking Database is generated whenever the position of the radar is specified using the terrain elevation data extracted from a SIMNET Database Interchange Specification (SDIS) database. The Terrain Masking Database contains a list of the sine of the elevation peaks for each radial, of four degree azimuth width, centered at the radar location. The sine of the elevation angle of the vehicle and its ground range are then checked against the appropriate radial from the Terrain Masking Database to determine if the vehicle is blocked by a closer terrain peak. If the vehicle is masked by terrain, no further processing is performed on it.

The total noise signal considered by the Generic Radar Model which interferes in the detection of a vehicle is composed of the radar receiver noise signal plus the signal due to jamming for a particular spherical sector plus the returned chaff signal. Since electronic countermeasures (ECM) and chaff are not included in the SIMNET specification, the current implementation only models radar receiver noise. Jamming and chaff are planned for inclusion in the DIS specification.

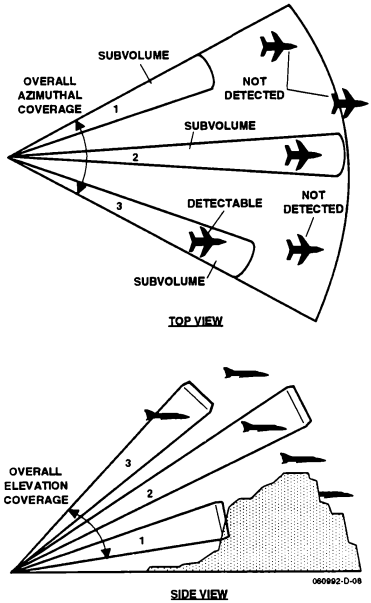


Figure 5. Specific Coverage Area

The Specific Coverage Area Check identifies whether a vehicle is within a search or track subvolume. A subvolume is a spherical sector described by minimum and maximum Range, Azimuth, and Elevation Angles (see Figure 5). The Azimuth Angles are specified relative to the boresight, or Primary Target Line (PTL) Azimuth Angle. The Elevation Angles are specified relative to the or PTL Elevation Angle. To generalize the search and track volume specifications for a radar, an unlimited number of subvolumes describing the volume is allowed. Each subvolume will have a Subvolume Reference ID so that individual subvolumes can be dynamically created, modified, and deleted.

For each radar, dynamic lists of the current track and search subvolumes are maintained. Each previously tracked vehicle is checked against the appropriate track subvolumes to determine if it is still within tracking space. If not, drop track processing is performed on the track. Untracked vehicles are checked against the appropriate search subvolumes to determine if they could be detected in this cycle.

One aspect of this processing is the determination of which vehicles are illuminated by the radar during this scan. All vehicles that are subjected to the coverage area check and that fall within the spherical sector for a search subvolume, ignoring the range limitations, are flagged as being illuminated. This data is required later for the generation of the Radiate PDU.

All vehicles that pass the Specific Coverage Area Check are subjected to Signal Strength Evaluation. A basic radar equation is utilized for computing the returned radar signal from a vehicle. The first step is to compute the Target Power Signal:

$$\text{Target\_Power\_Signal} = (\text{Power} * \text{Gain}^2 * \text{Attenuation}^2 * \text{Speed\_of\_Light}^2 * \text{Radar\_Cross\_Section}) \,/\, (4\pi^3 * \text{Target\_Slant\_Range}^4 * \text{Frequency}^2)$$

The Generic Radar Model requires a Radar Cross Section for a particular vehicle in determining the returned signal strength. Actual radar cross sections of vehicles depend on frequency and vary with small incidence angle changes. Accurate modeling of radar cross section requires huge databases indexed by radar, vehicle, and frequency for each roll, pitch, and yaw incidence angle. Such a database was considered beyond the scope of this project. Instead, a coarse approximation of a vehicle's radar cross section, allowing for some incidence angle variation, was developed. The radar cross section is obtained from a Vehicle Data File which contains records for individual vehicle types.

Each vehicle entry contains an approximate radar cross section for several Z-axis angles. The Z-axis of a vehicle is perpendicular to the plane of its top or bottom. Although the X-axis and Y-axis aspect angles also affect the radar cross section, the coarse approximation only takes into account the radar to Z-axis aspect angle at 30 degree increments.

Next, the Target Power Signal is compared to the radar receiver noise. If the Target Power Signal is greater than the noise then the vehicle is flagged as detected.

One or more Radiate PDUs must be generated after completion of this scan. All vehicles that have been "detected" are included first in one or more Radiate PDUs. Then, all vehicles that were illuminated but not detected are listed.

GENERIC MISSILE MODEL

As with the Radar Investigation, an investigation was undertaken to determine the availability of a suitable generic missile model, either in code or algorithm. At the outset, it was established that real-time modeling of multiple missiles against a potentially large number of targets was critical. Missile algorithms that modeled the detailed fin action or provided 6 degrees of freedom (6-DOF) were considered inappropriate because they are both computationally intensive and beyond the fidelity required by the visual simulations of SIMNET/DIS. The investigation encompassed both a library search and an internal review of available missile models documented in Sanders' Modeling Lab.

Two "generic" missile models were reviewed which included code for the entire engagement problem. This included radar detection, launcher selection and motion, aerodynamics, guidance, fuzing, seeker, system response times and reliability, trajectory, ECM, and missile vulnerability. These models could not be used for two principle reasons - first, the modeling of the processes was more detailed than desired, and second, much of the processing is performed by other parts of the Protocol Converter, or is not designed to be part of the PC at all. For example, the ADT is responsible for launcher selection and motion, system response times, and other system specific details. Additionally, it was determined that modeling of the many types of missile target acquisition methods such as navigational tracking, passive/semi-active/active radar, Home-on-Jammer, Track-via-Missile, IR, and visual is beyond the intentions for the SPPC's Generic Missile Model. The end result of the investigation was that no internal missile model was found that could directly be used for the SPPC. The existing SIMNET 6.6.1 Semi-Automated Force (SAFOR) Missile Model source code was then investigated. This software is written in 'C' and is tightly coupled to the other SAFOR modules, which prohibited its direct reuse. Many of the algorithms, however, were used in the basic design of the SPPC Generic Missile Model design.

The characteristics of the modeled missile are input at the MMI; these include the maximum tracking angle, fuel, maximum turning rate, maximum speed, fusing distance, as well as the warhead and detonation types. The ADT is responsible for providing dynamic information to the protocol converter, such as the selected launcher and designated target. The ADT can also abort the missile at any time during the flight.

The Missile Model deals with the missile from launch until explosion or ground impact. The Missile Flyout portion of the Missile Model is designed as a state machine. The four states of Missile Flyout are illustrated in Figure 6. At launch, missile flyout modeling begins and the missile is in the "Tracking" state. If it reaches Fusing Distance to the target, the missile is then "Armed". If the target falls outside of the Maximum Tracking Angle or if the vehicle has been dropped then the missile is "Lost". Finally, once the missile runs out of fuel, it begins a free fall and is "Crashing".

Missile Launch

The missile is launched as a result of an operator action in the trainer/simulator. The modeling of the missile is based on the kind of missile launched and corresponding characteristics previously input at the MMI.

Missile Flyout

The Missile Model flies the missile after the target using the basic laws of motion, ignoring details like friction and lift. The missile exits the tube in the direction of the launcher at its Initial Speed. The missile will accelerate at a constant rate up to a Maximum Speed. As long as the missile keeps the

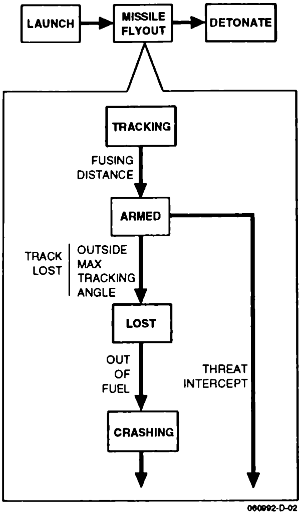


Figure 6. Missile Model State Transition Diagram

target within the Maximum Tracking Angle, it will attempt to match the tangential velocity of the target, to keep it in view. The remainder of the available speed is used for closing in on the target. There is no modeling of the target detection process except for the angular check.

Missile maneuvering is limited by a Maximum Turn Rate capability, measured between the original velocity vector and the updated velocity vector. If the missile loses track of the target, it will fly straight until its Fuel Time is expended. Once fuel is expended, the missile begins a free fall with its XY velocity components remaining the same and its Z continuously changing due to gravity, until impact with the ground (no detonation).

Missile Detonation

Once the missile is armed, it is checked for passing the point of closest approach, i.e. the range rate begins increasing. When this occurs, the missile is presumed detonated at the closest point. The missile may also be prematurely detonated by an ADT message.

The Missile Model is completely responsible for generating and issuing the required SIMNET messages. It will issue the Fire PDU, multiple Vehicle Appearance PDUs, Impact PDU, and the Deactivate PDU.

CONCLUSIONS

The SPPC program has confirmed that the protocol converter approach to linking an existing air defense trainer/simulator to a simulation network is a viable technical solution.

Using the Protocol Converter, the PATRIOT OTT has been successfully integrated with other vehicles playing on SIMNET. That is, a PATRIOT OTT operator has successfully engaged multiple tracks (aircraft SAFORs), maintained by the detection model, with multiple missiles flown by the Missile Model. As of the date of this paper testing of the PC is still in process and the upper limits (number of tracks and targets) have not been determined.

The protocol converter has been designed to be configurable for use with different air defense trainers. Customization can be in the form of replacing entire generic models, such as the Detection Model, and by using the current set of configuration parameters provided through the MMI to modify the processing. Use of the protocol converter to participate on SIMNET now requires that a trainer is modified only to implement the generic interface, which is far less complicated than the complete set of SIMNET protocols. These elements have maximized the intelligence and processing required by the Protocol Converter itself.

By locating the protocol converter on the SIMNET LAN and connecting it to the trainer via a low speed WAN, we have been able to reduce the network bandwidth requirement and associated delays.

Upgrading the protocol converter from SIMNET to DIS version 1.0 will not require major modifications due to the similarities in the protocols. The differences between SIMNET and DIS have been investigated concurrently with the development of the PC, and the conversion task is presently in process.

ABOUT THE AUTHORS

Daniel A. Bradford received his Bachelor's degree in Computer Engineering Technology from Northeastern University in 1989. He has been involved in the planning and execution of various IR&D programs to expand the interoperability of training systems at Lockheed Sanders. He is presently the Principle Investigator of the SIMNET/PATRIOT Protocol Converter Research and Development Program upon which this paper is based. Mr. Bradford is a member of the Tau Alpha Pi Honor Society.

Danielle M. Eriksen is a Senior Software Engineer in Lockheed Sanders' Information Systems Division. Ms. Eriksen received her Bachelor of Science degree in Computer Science from the University of Lowell in 1985. She has been involved in software development on several major programs within Sanders, including a real-time simulation of an airborne electronic countermeasures system.

Huat K. Ng has received a Bachelor of Science degree in Computer Engineering and an Master of Science degree in Electrical Engineering from the University of Central Florida in 1986 and 1989 respectively. He has been working with the Institute for Simulation and Training since 1990, and his work is primarily focused on real-time simulation networking. He has been involved with Distributed Interactive Simulation (DIS) Standard development and implementation of application level gateways to interface dissimilar simulation protocols.

Alan M. Thibodeau is a Senior Principal Software Engineer in Lockheed Sanders' Information Systems Division. He received his Bachelor of Science Degree in Physics from Lowell Technological Institute in 1974 and his Master's Degree in Computer Engineering from the University of Lowell in 1979. He has been involved on several large trainer system projects, most recently on the PATRIOT Operator Tactics Trainer (OTT). He is presently assigned as the Program Engineering Manager for Air Defense Trainer related projects.

LIST OF ACRONYMS

ADT	Air Defense Trainer
DARPA	Defense Advanced Research Projects Agency
DIS	Distributed Interactive Simulation
DOF	Degrees of Freedom
ECM	Electronic Countermeasures
FP	Fire Platoon
IR	Infrared
LAN	Local Area Network
MMI	Man Machine Interface
OTT	Operator Tactics Trainer
PC	Protocol Converter
PDU	Protocol Data Unit
PTL	Primary Target Line
SAFOR	Semi-Automated Forces
SDIS	SIMNET Database Interchange Specification
SIMNET	SIMulation NETwork
SPPC	SIMNET/PATRIOT Protocol Converter
STRICOM	Simulation, Training, and Instrumentation Command
WAN	Wide Area Network

AIAA-92-4156-CP

DESIGN AND PERFORMANCE OF THE CENTRIFUGE - BASED DYNAMIC FLIGHT SIMULATOR

Jacob Eyth, Jr.* and Peggy L. Heffner
Naval Air Warfare Center, Aircraft Division, Warminster
Warminster, Pennsylvania

Abstract

The Dynamic Flight Simulator located at the Naval Air Warfare Center, Aircraft Division, Warminster has demonstrated a unique capability to perform motion-based flight simulation over the years. Using a 50-foot human centrifuge as its motion platform, it bridges the gap from fixed-based flight simulators to flight testing by creating the actual stresses of flight on the pilot. Under these realistic conditions, simple tasks become difficult and decision making can often be delayed. By using the Dynamic Flight Simulator, operational deficiencies can be identified early in the development program before they are uncovered in flight tests. This approach can result in significantly lower development costs.

This paper will explain the overall design of the Dynamic Flight Simulator and describes several successful test programs in which it has been used. Applications range from early NASA astronaut training to the current high angle of attack (HAOA) and thrust vectored aircraft studies. Also included in the paper are discussions of future applications and upgrades intended for the facility.

Equipment Configuration

General Description . The Dynamic Flight Simulator (DFS) features a 50 foot arm centrifuge with a two axis gimbaled gondola, which is driven by a 16,000 horsepower main drive motor. Aircraft cockpits are inserted in the gondola to create the motion based flight simulator.

The facility is comprised of 5 general areas: (1) the centrifuge device, (2) the cockpits, (3) the simulation computer system (which provides aircraft model inputs to the cockpit and centrifuge control system), (4) the centrifuge control system (provides inputs to the centrifuge motor and gimbal system), and (5) the flight deck (provides access to the gondola and medical monitoring facilities for the pilot).

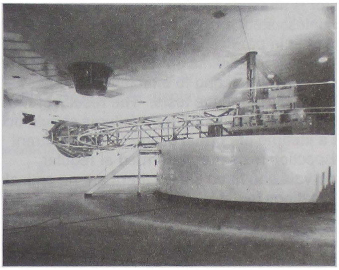


Fig. 1. Naval Air Warfare Center Centrifuge

The main portions of the facility are situated on three floors of the Centrifuge Building (see Figure 2).

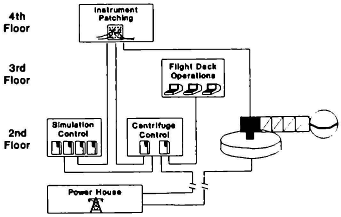


Fig. 2. Operating System Layout

----

[*] Branch Head, Crew Systems Facilities
Senior Member AIAA

The Centrifuge Device. The main housing for the centrifuge is a cylindrical, copper shielded, reinforced steel and concrete building 130 feet in diameter and approximately three stories. The chamber which houses the centrifuge itself is 30 feet high.

The centrifuge motion base is capable of attaining up to 40 g's with a 1000 pound payload. Because of the torque available from the main motor, G-onsets of up to 13 g/sec are attainable.

Drive Motor. In the center of the 110 foot operating floor is a 180 ton, 4000 horsepower, vertically mounted, direct drive d-c motor, built by the General Electric Company. This motor has a peak rating of 16,000 horsepower, and a nominal rating of 4,000 horsepower. An 80 ton rotor is attached to a 50 foot Armco 17-4 PH stainless steel arm. Considerable effort was exerted in perfecting welding techniques and in designing weld joints on test specimens of the arm and gimbal ring to enable them to pass 10,000 cycle fatigue tests. The results of these tests indicated that no weld failures should occur in a lifetime of normal usage. Dye penetrant inspection tests are performed at regular intervals and have shown there are no critical indications to date.

Centrifuge Arm Variations. The arm is capable of having a 22 foot section removed in order to provide a shorter radius centrifuge for increased tangential accelerations and increased rate of G onsets. With this section and the gondola removed the remaining 22 foot section is structurally capable of supporting a 5000 pound payload up to 100G. The 100G capability can be provided by making certain alterations in the control system which would allow the main accelerator motor to be operated at a maximum speed of 110 rpm compared to its present maximum speed of 48.5 rpm.

Sliprings. Slipring provisions in the gondola for air-conditioning/vacuum, hydraulic oil, and pneumatics have also been included in the design of the centrifuge.

Data is transmitted to the first and second gondola axes and to the instrumentation via a total of 124 electrical sliprings. A comprehensive study was made to determine the best ring and brush finishes to use for low noise (1 micro volt) circuits which would yield good working life. The combination found most suitable was rhodium-plated copper rings with 6 micro inch finish, and gold-plated wire brushes.

Gimbal System. The DFS gondola is suspended in a dual gimbal system. The outer gimbal, or roll gimbal is capable of a full 360° continuous rotation, which is the same as that of the inner or pitch gimbal. In DFS mode, the gimbals operating range is from -90° to + 90°. Engineering provisions have been made for the insertion of a third axis or yaw gimbal capsule.

Vacuum Caps. Vacuum caps are available for installation on the gondola for simulating high altitude flight. These caps have been tested for safe operation to the vacuum equivalent of 60,000 feet. However, the caps are designed to operate to 100,000 feet.

Cockpits. The DFS can be configured with any of three available cockpits: the F-14D/F-14A, the Lightweight Generic, and the Advanced Technology Demonstrator Supine Seat.

When not in the DFS, the cockpits are mounted in a ring structure on the ground. While mounted, the ring structure may act as a ground station for development purposes.

F-14D/A Cockpit. The F-14D/F-14A cockpit is a high fidelity cockpit which contains programmable multifunction displays, a Head-Up-Display (HUD), cockpit instruments, consoles, throttles, and an electrohydraulic stick/rudder control loader. The entire crewstation weighs approximately 2500 pounds which limits its maximum G capability to 10 g and the maximum G-onset to 4 g per second.

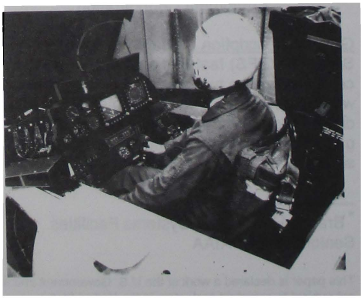


Fig. 3. F-14D Cockpit

Lightweight Generic Cockpit. The Lightweight cockpit is presently being used for Thrust Vectored Aircraft studies. This cockpit includes a fighter aircraft style ejection seat, an electronic sidearm controller, a three channel visual display, and a head down 9-inch CRT display. It weighs approximately 1350 pounds and is capable of a 15 g maximum and a g-onset of 8 g per second.

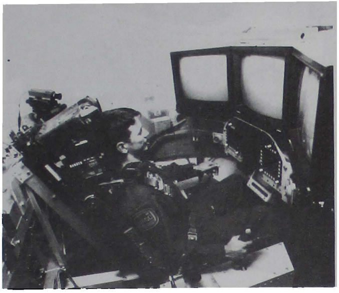


Fig. 4. Lightweight Generic Cockpit

Advanced Technology Demonstrator Cockpit. A third cockpit, the Advanced Technology Demonstrator Supine Seat is under development. It is comprised of a supine seat, a sidearm controller, and three flat panel head-down displays.

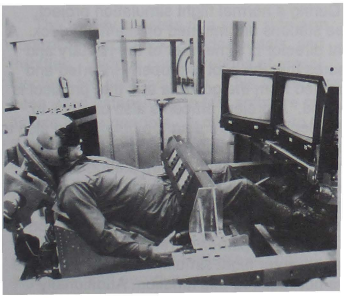


Fig. 5. Advanced Technology Demonstrator Supine Seat Cockpit

Simulation Computer System. The simulation computer system has been developed to provide the aircraft-like environment. It is comprised of three components: the simulation computer, the display processor, and the visual system. This system receives cockpit inputs, processes them, and provides updated outputs to the displays and the control system.

Simulation Computer. The simulation computer, an Encore 32/6750 mainframe, is used for simulating the aerodynamic model. The Encore operates in real-time with a frame time of 50 milliseconds. This computer is also used for collecting test data on magnetic tape for off-line reduction and analysis.

Display Generators. The display processing is accomplished by a Gaertner Inrad II digital programmable display processor. This is primarily used for the head down display generation for the F-14D, F-14A and ATD Supine Seat cockpits. It creates the attitude directional indicator, raster output, RGB, NTSC display. A Silicon Graphics computer is being considered as an upgrade for the Gaertner system. This will allow for an easily reconfigurable cockpit display.

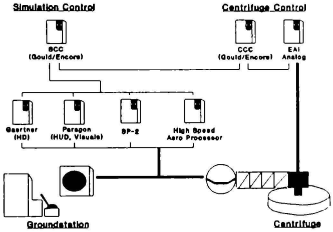


Fig. 6. Interface Diagram

Visual Systems. All three cockpits can be used with one of the two available out-the-window display systems: a 3 channel high resolution Paragon visual display or a single channel Rediffusion SP-2.

The Paragon system uses three high resolution monitors which provides the pilot with a 30^{o} x 120^{o} field of view. This system has the capability of projecting Head Up Display (HUD) symbology superimposed on the real-world scene. Since the HUD is programmed into the software, there is no requirement for an additional display processor.

The second visual display system is a Rediffusion SP-2 system. This system uses a single window virtual image projection system and provides the pilot with a limited 32° x 48° field of view. Because the SP-2 uses virtual image projection, it can be used in conjunction with an actual HUD and combiner lens.

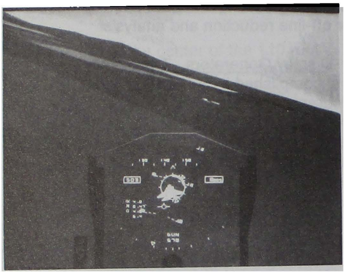


Fig. 7. Visuals Photo

Centrifuge Control System. The centrifuge control system receives the aerodynamic data inputs from the simulation system via an analog to digital interface and provides updated control commands to the gimbals and the centrifuge arm. The gimbal drive algorithms resident in this portion of the system provide the coordinated motion cueing which produces the sensation of flight.

The centrifuge control system is comprised of an Encore 32/6750 digital computer, and an EAI 2000 analog computer. The digital computer contains the centrifuge control algorithm that calculates the Gx, Gy, and Gz components based on the aircraft performance data it receives from the simulation computer. These outputs are then passed to the analog computer which in turn transforms the commands into analog signals and drives the centrifuge.

As the pilot moves the stick, the stick positions are transferred to the simulation control computer which then calculates the new aircraft orientation. The display information is also recalculated to denote the change. The new aircraft position information is then transferred to the control computer which calculates the g components. The g components are then converted into analog signals and passed on to the gimbals of the gondola. The pilot senses the commanded changes in the orientation of the gondola and along with visual system feedback attains the sensation of flight.

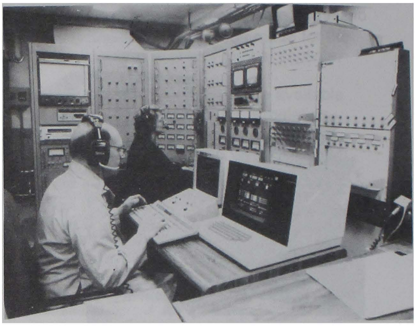


Fig. 8. Centrifuge Control Room

Flight Deck. The flight deck is used for monitoring the medical safety of the pilot and the progress of the experiment via instrumentation, camera monitors, and other medical monitoring equipment.

During a normal flight simulation project, three stations are manned on the flight deck. A flight director is a member of the military and ensures that the pilot is properly inserted and extracted from the gondola and that protocol is followed. A project officer is also on deck to make decisions regarding the project and also to ensure that protocol is followed. The flight surgeon is located at the third station and by monitoring the medical equipment and cameras determines if the subject is safe based on the biomedical feedback. There is also a station for the medical corpsmen to record physiological data, however flight simulation projects to date have not required this function. Additionally medical corpsmen are standing by in the unlikely event of a medical emergency.

The pilot, who is usually fitted with EKG leads and other various medical monitoring equipment, enters the gondola via the flight deck.

The flight deck also contains equipment to record various data including, medical, engineering, and video with voice recordings. This data is monitored and analyzed real-time to ensure the safety of the pilot as well as progress of the experiment. Since the data is being recorded, it may be more thoroughly analyzed at a later time.

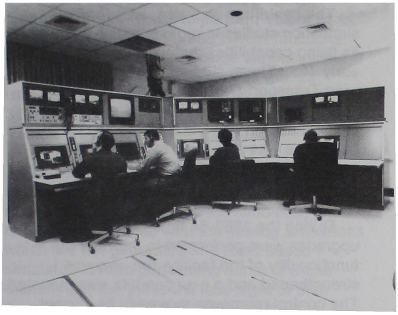


Fig. 9. Flight Deck

Experimental Control Station (ECS). An Experimental Control Station may be used within the system as a replacement for the cockpit. It is mainly used as a software development and test facility.

The facility consists of a seat, side-arm controller (force stick), rudder pedals, and throttles. Various configurations of visual displays are also available. This station has been used extensively to test software prior to its use in the gondola. A pilot is able to test the simulation software by flying the system in a stand-alone mode. The facility can also be configured as remote centrifuge cockpit controls in which case it is used to provide the inputs to the centrifuge without subjecting a human to the effects. This provides the capability of preliminarily testing the effects of control algorithm and aerodynamic software modifications and its relation to the stresses on the gimbals and the centrifuge without exposing humans to unpredictable results.

Static training for the pilots prior to centrifuge flights is also provided at this facility.

Simulation Software

Aircraft. The simulated aircraft models which are currently available are a Generic Aircraft Model (based on an F-18), F-14D, A-6, and a fluidic version of an F-18. These models are resident on the Encore 32/6750 simulation computer.

The Generic model does include thrust-vectoring capabilities. This model has been used with a side-stick controller and was enhanced in an effort to provide Level 1 flying qualities. This model has been modified to increase drag in order to prevent exceeding 1.0 Mach so as to maintain control surface stability due to high gain feedbacks. Further optimization relative to Mach number may be performed to prevent this characteristic and to simulate supersonic flight operations.

The fluidic F-18, F-14D, and an A-6 model are also available but have not been thoroughly tested in the centrifuge environment.

Under investigation is the feasibility of performing helicopter studies on the centrifuge. If so, a helicopter model based on an SH-60 helicopter may be added to the aircraft models.

Control Algorithm. The control algorithm being used is resident on the Encore 32/6750 control computer. Aircraft parameters, Gx, Gy, Gz, roll, and pitch are input into the algorithm, and after the computation is complete the resulting gimbal and arm commands are output. The processing is completed at a 50 millisecond frame time. Various parameters may be modified including limits, filters, constants, gains and output data definition (data which is recorded) to provide a better simulation.

Data Collection and Analysis.

Strip Charts. Strip charts are available for recording 16 channels of aerodynamic data, 16 channels of centrifuge control data, and 16 channels are available for flight deck monitoring and recording a combination of the control and aerodynamic data. Additionally 8 channels of medical data may also be recorded which may include EKGs, G-Suit, and various G information which the subject may be experiencing.

Video. A VHS video recorder with voice overlay is available for recording the camera monitors during the run.

Analog Data. Analog data may be acquired on a 14 channel TEAC recorder. The storage medium is a VHS tape.

Digital Data Acquisition. A data acquisition computer is available on the flight deck for acquiring and recording up to 13 channels of data on computer disk for further analysis off-line.

Both Encore computers (the simulation computer and the centrifuge control computer) are also equipped with the ability to record data on magnetic tape. This data consists of variables used within the programs. The centrifuge control computer and the aerodynamic simulation computer are capable of recording up to 99 variables at a maximum rate of every 50 ms. This data is downloaded onto a PC and graphed and analyzed in order to compare the flight characteristics to the centrifuge commands to insure proper operations and performance. It may also be used as inputs into control algorithm software models to compare the results with respect to modifications and new designs of the existing algorithm.

Historical Background

The centrifuge was designed and built during the period of 1945 until 1952. On June 17, 1952 the Aviation Medical Acceleration Laboratory at the U.S. Naval Air Development Center, Johnsville, Pennsylvania was dedicated. Johnsville was later renamed Warminster and the Aviation Medical Acceleration Laboratory became departments within the Naval Air Development Center some of which study acceleration effects and others who perform flight simulation.

Astronaut Training. As space flight became a reality, studies pertaining to the effects of acceleration on the body were performed at the centrifuge. Beginning in 1959, Mercury astronauts were trained at the centrifuge to familiarize themselves with G-forces and other various aspects of space flight as did the Gemini and Apollo astronauts. The early acceleration studies which were performed used what appeared by today's standards to be "medieval" techniques and procedures. Findings of these studies showed that tasks which are designed at 1 g are not always easy to perform at higher g levels. As a result, in 1974, Space Shuttle astronaut trainees performed reach and visibility studies for their new cockpit. These findings are even more essential in the complex aircraft cockpits of today.

The centrifuge was also used for flight simulation studies in regards to the aircraft and its flying capabilities during the late 1960s and early 70s. These studies included the Swept-wing Transport (720-B) Clear Air Turbulence study, F-4 Spin study, F-14 Buffet and Spin studies, and an A-7 Catapult study. The centrifuge was also used for various human factors studies including seat and gear evaluations during this time period.

During the 1963-64 time frame, major upgrades were performed to improve the functionality of the facility, the arm was strengthened and a new gondola was mounted. The control room was moved to its current location and modernized. Much of this equipment is still in use today. Some computer modernization has occurred over the years but there have been no significant upgrades to the arm and gimbals since the 1960s.

Early F-14 Work. In 1984 the DFS came on-line. In this mode, the centrifuge is used solely for flight simulation. One of the first projects which occurred in the DFS was an F-14 Spin project whereby the pilots were trained in spin recognition. As a result a spin arrow was designed and tested, and now appears in F-14D aircraft. Additionally it was found that pilots were not locking their restraints. When flying under significant Gs, they were pulled forward and were unable to read their instruments and recover from the spin. Occasionally it becomes very difficult to reach the ejection pull under these conditions. Locking restraint mechanisms were designed and are now being incorporated in the latest aircraft seats (NACES) being produced.

Enhanced Fighter Maneuverability (EFM). In 1986, the centrifuge was used for a preliminary EFM investigation. Using an F-14D cockpit, and modified F-14 software, the centrifuge was able to provide the flight characteristics of High Angle of Attack (HAOA) aircraft. This type of study is being investigated using a lighter weight cockpit and modified F-18 software in order to provide more of the yawing cues, essential in HAOA aircraft.

Man-in-the Loop Applications

Recent Applications. Presently the DFS is being used to ascertain the capabilities of using the Lightweight Generic Cockpit and the Generic Aircraft Model to perform Thrust Vectored Aircraft (TVA) and HAOA functions. These applications can include various display generations and modifications, as well as training pilots who will fly aircraft such as the X-31 and F-18 High Angle-of-Attack Research Vehicle (HARV).

The sensations which are felt in this type of aircraft are unusual. By performing the TVA maneuvers in a motion-based simulator, pilots can become familiar with the yawing sensations which are not generated in a fixed based simulator. When performing some of these maneuvers, the pilots will lose sight of their opponents and will need to rely on their displays. Displays which will assist them in this situation can be easily modified in the DFS simulation environment in order to create the optimum display for their use. Additionally, their workloads in performing these tasks can be measured and compared when using the various display configurations. This saves flight time and reduces risk as does all simulation, however, the DFS provides an environment which is one step closer to the aircraft by providing the g forces.

Future Applications. With new tactics such as HAOA maneuvers, it is becoming more evident that the pilots will need to become more reliant on advanced cockpit displays to augment their visual out-the-window reference. This is also true with various air to ground applications. In certain high angle-of-attack situations, pilots lose sight of their opponents, be it aircraft, surface to air missiles, etc. and need to rely on the displays to track these opponents. This will most likely change the piloting techniques used in the cockpit and will also change the heads up and heads down displays which have traditionally been the primary flight instruments. The centrifuge is a safe and realistic laboratory in which to verify these modifications.

Nap of the Earth Flight. Nap of the earth flight is also a very dangerous flight regime which requires aggressive maneuvers resulting in high G situations. A flight scenario such as the A-10 would typically fly close to the ground, dropping ordnance and pulling up at high angles of attack. This not only obscures the pilot's vision but also creates a significant amount of Gs on the pilot. Training for this type of flight regime are under investigation.

Pilot's Associate. Artificial Intelligence systems are also receiving a lot of attention and are being considered for testing in the centrifuge prior to costly aircraft flight time. The centrifuge would provide a prime testing ground for these types of devices. The units can be integrated into the centrifuge system and thoroughly tested prior to installing it on an aircraft.

Helicopters. Helicopters are being designed to withstand more gs and perform more aggressive maneuvers. Future helicopter are becoming highly maneuverable and will be designed for 3-5 gs. Since higher g-forces are a new flight sensation which helicopter pilots are unfamiliar with, training in the centrifuge may be desirable.

Future Facility Upgrades

State of the Art Visual System. A new state of the art visual system, which incorporates texturing, faster processing, and ability to display more reference points is being considered for upgrade.

Reconfigurable Cockpit (Glass Cockpit) The latest trend in flight simulation involves building one cockpit and allowing it to be easily reconfigured in order to simulate any cockpit desired. Since it takes nearly three to fourteen days to change cockpits (depending on the installation), and then perform structural testing to insure the safety of the installation this is considered a highly desirable approach for the centrifuge.

A reconfigurable cockpit which provides both a sidearm controller and a center stick, a three channel out-the-window display with a HUD overlay, and a CRT which can be quickly reprogrammed into any aircraft head down or experimental configuration, is shown below. This does not provide the high fidelity that an individual aircraft cockpit mock-up provides however, it does provide quick reconfiguration time.

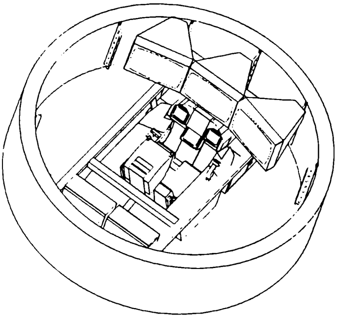


Fig. 10. Reconfigurable Cockpit

Modular Approach to Hosting Aircraft Models. Existing simulation software is designed to host only the aircraft models which have been developed for our facility. A more modular approach and restructuring of the software is being considered, whereby, outside customers would be able to host their validated aircraft models. All future upgrades are being made with this approach in mind.

Conclusion

The DFS is a highly sophisticated flight simulator, capable of generating motion characteristics not found in other types of simulators. In reading many magazine articles, and talking to hundreds of pilots, the same observation is made, without the sustained motion cues, a fixed based, or minimal motion based simulator "can feel like flying a video game." As stated many times before, flight simulation is still a cost effective alternative to actual aircraft flight training. The centrifuge flight simulation is the stepping stone between a fixed based simulator and the real aircraft.

References

1. "Dedication of the Aviation Medical Acceleration Laboratory," _ U.S. Naval Air Development Center, Johnsville, PA, 17 June 1952.

2. Crosbie, R.J., "The Centrifuge Method of Dynamic Flight Simulation," Aerospace Medical Research Department Report, undtd.

3. Dornheim, M. March 11, 1991. X-31 Flight Tests to Explore Combat Agility to 70 Deg. AOA. Aviation Week & Space Technology.

4. Bischoff, D. E., Poole, D. A., and Eyth, Jr., J. "Flight Simulation Fidelity in a Total G-Force Environment." AIAA Paper 85-1730, Flight Simulation Conference, July 1985.

5. Crosbie, R. J. and Kiefer, D. A., "Controlling the Human Centrifuge as a Force and Motion Platform for the Dynamic Flight Simulator." AIAA Paper 85-1742, Flight Simulation Conference, July 1985.

AIAA-92-4157-CP

DEVELOPMENTAL EVALUATION OF A CENTRIFUGE FLIGHT SIMULATOR AS AN ENHANCED MANEUVERABILITY FLYING QUALITIES TOOL

Dennis A. Kiefer* and Jeffrey F. Calvert
Air Vehicle and Crew Systems Technology Department
Naval Air Warfare Center, Aircraft Division
Warminster, PA

Abstract

Significant improvements in high angle of attack aerodynamics, flight control systems and the use of thrust vectoring is providing current and anticipated aircraft with enhanced maneuverability. These aircraft generate unusual rates and accelerations which will severely affect pilot spatial orientation and situational awareness during air combat maneuvering. The developing technology of centrifuge flight simulation offers the prospect of ground-based flying qualities and human factors testing in this same environment.

Centrifuge flight simulation technology, as implemented on the Dynamic Flight Simulator, has shown its value with F-14 flat spin and mishap investigations, preliminary enhanced fighter maneuverability studies, and physiological investigations in a realistic flight environment. Recent and current efforts are to expand this role for potential use as an enhanced maneuverability simulation tool. Specifically targeted for evaluation is the ability to perform piloted analyses of critical displays during high angle of attack enhanced maneuvering tasks.

Improvements have been made to the Dynamic Flight Simulator motion base control and actuator quality, cockpit displays, data collection capability, and compatible tactical aircraft models. The evaluation involved analyzing improvements to motion fidelity and demonstrating the potential for addressing a broader class of aircraft research, development, test and evaluation issues. Limitations are separated into those inherent to the technology and those dependent on the Dynamic Flight Simulator implementation. Tradeoffs between control method and mission applications are shown.

Nomenclature

AM	- Aircraft Model
AOA	- Angle of Attack
A	- A-Gimbal Angle (+ Left/Inward)
Ar	- Normal Acceleration
At	- Radial Acceleration
B	- B-Gimbal Angle (+ Up)
CCA	- Centrifuge Control Algorithm
CFS	- Centrifuge Flight Simulation
C_{n_β}	- Static Directional Stability Coefficient
DFS	- Dynamic Flight Simulator
DOF	- Degrees of Freedom
EM	- Enhanced Maneuverability
FOV	- Field of View
G	- Translational Acceleration Felt at the Cockpit Station
HAOA	- High Angle of Attack
HARV	- High Angle Research Vehicle
KCAS	- Knots Calibrated Airspeed
LWC	- Light Weight Cockpit
MC	- Motion Controller
MS	- Motion System
P	- Roll Rate (+ Right Wing Down)
PSEM	- Pilot Sensory Estimate Model
PST	- Post-Stall Maneuvering
Q	- Pitch Rate (+ Nose Up)
R	- Yaw Rate (+ Nose Right)
TV	- Thrust Vectoring
$(\bar{\ })$	- Aircraft Parameter Pre-Limited at CCA Input
α	- Angle of Attack
β	- Angle of Sideslip (+ Nose Left)
δ	- Cockpit Control Position Reference
Φ	- Roll Attitude (+ Right Wing Down)
Θ	- Pitch Attitude (+ Nose Up)
ψ	- Heading Attitude
ω	- DFS Rotation Rate (+ Counter Clockwise)
Ω_A	- Angle to Vertical Due to Angular Accelerations
Ω_V	- Angle to Vertical Due to G Vector

Subscripts

a	- Pertaining to Aircraft
ac	- Aircraft Variable Input to the CCA
b	- Referenced to Aircraft Body Axes

* Member AIAA

c	- Pertaining to DFS Parameters or Command Parameters
cm	- Gimbal Angle Command Including Effects of Servo Dynamics
lat	- Pertaining to Lateral Control Input (+ Right)
long	- Pertaining to Longitudinal Control Input (+ Pull)
ped	- Pertaining to Directional Control Input (+ Right)
r	- Pertaining to Radial (+ Inward)
rv	- Resultant of Centrifuge Radial and Vertical Acceleration (g)
t	- Pertaining to Tangential (+ Forward)
th	- Pertaining to Throttle
x	- Pertaining to x-axis (+ Forward)
y	- Pertaining to y-axis (+ Right)
z	- Pertaining to z-axis (+ Down; Gz + Up)
_FEB	- Pertaining to Servo or CCA Parameter Feedback

Introduction

Enhanced Maneuverability (EM) has been recently made possible primarily through the use of thrust vectoring (TV). TV provides the additional controllability required for maneuvering at flight conditions beyond stall angles of attack by using engine thrust to provide control moments much the same way control surfaces provide aerodynamic control moments. Aircraft such as the X-31A and the F-18 HARV are currently demonstrating controllable EM capability at high angles of attack (HAOA) where the unusual combinations of rates and accelerations are expected to be very disorienting. Recent studies have shown that optimized displays integrating aircraft performance, handling qualities, and situational awareness will likely be required to allow the pilot to take full advantage of EM capability during tactical operations [1,2].

From 1985 through 1987 the Dynamic Flight Simulator (DFS) located at the Naval Air Warfare Center, Aircraft Division, Warminster, was evaluated for use as a high angle of attack EM centrifuge simulation tool. The advantage that the DFS can provide over fixed based simulation and conventional motion base simulation is the ability to provide sustained motion cues. However, these investigations identified some improvements were necessary to fully optimize sustained motion fidelity.

In response, since 1987 several modifications to the DFS have occurred in both the control algorithm and the mechanical configuration. These changes were engineered to dramatically improve the overall motion fidelity. Between September 1991 and April 1992, developmental testing and evaluation of the DFS motion algorithm has continued. To date several key improvements have been documented, including improved lateral acceleration cues, roll response cues, and improved rapid maneuvering response characteristics without excitation of unwanted structural modes. Results indicate that the DFS is capable of providing the sustained motion fidelity necessary to conduct limited HAOA flying qualities and human factors investigations. This report covers the RDT&E efforts to date which are intended to fully develop, test, and document the DFS to the level necessary for utilization as a credible HAOA research motion flight simulator.

I. BACKGROUND

Centrifuge Flight Simulation

The experience of flight is most authentically created in the aircraft. The reasons for simulating the flight environment are reasons of practicality. The economy, safety, repeatability, and flexibility of ground-based flight simulation often make it the best or only approach to obtain needed results. Each simulation technology provides an environment different from actual flight. Simulation method selection is based upon the task to be performed. The method should reproduce conditions significant to the task while maintaining acceptable differences for those less important. Centrifuge flight simulation (CFS) technology provides a simulation solution to handling qualities and human factors tasks which require a more representative motion environment than available in other ground based simulation techniques.

DFS Experience

In late 1986, an enhanced maneuverability (EM) study was conducted on the Dynamic Flight Simulator (DFS) located at the Naval Air Warfare Center, Aircraft Division, Warminster, Pennsylvania. The purpose of the study was to assess post-stall (PST) handling qualities, control law authority, cockpit displays, and the effects of unusual rates and accelerations associated with PST maneuvering on pilot workload and mission performance Approximately 23 hours of man-in-the-loop simulation was conducted over a seven day period using three project pilots. The simulation was comprised of an F-14A+ aircraft model configured with thrust vectoring capability and an aerodynamic database modified to allow controllable maneuvering up through 90 deg AOA. The actual simulation cockpit was retrofitted to resemble a modified F-14A. The single monitor visual display used a "light dome" grid superimposed on the sky to overcome anomalies associated with maneuvering with a limited field of view (FOV) in an effort to aid the pilot during target acquisition tasks at HAOA.

While test results showed the DFS provided adequate cues for pilot perceived axial (G_X) and positive normal acceleration cues ($+G_Z$), several drawbacks were identified. Angular acceleration response cues were perceived as low. Adequate lateral acceleration (G_y) and negative normal acceleration ($-G_Z$) cues were identified as necessary to realistically simulate HAOA maneuvering. Finally, rapid maneuvering often resulted in excitation of a structural mode causing uncommanded oscillatory vertical motions (defined as the hobby-horse mode); this mode was unrealistic of an actual flight environment and considered a nuisance by the evaluation pilots.

Since the F-14 EM study, several modifications have been made to DFS hardware and software specifically to address the identified anomalies. Table I outlines these modifications along with the targeted improvements.

TABLE I
DFS IMPROVEMENTS SINCE THE
F-14 EFM PROGRAM (1986)

MODIFICATION	IMPROVEMENT
Rehosted Aircraft Model on Local On-Site Computer	Fewer interruptions; More Consistent Response Times
Installed Lightweight Cockpit	Improved Response Capability
Moved C.G. Position closer to Center of Rotation	Reduced Susceptibility to Vertical Arm Oscillation Mode ("hobby-horse mode")
Implemented Improved Display System	Wider Field of View
Implemented Centrifuge Control Algorithm on Digital Computer	Improved the Quality of Control Filters
Modified Gimbal Gear Drive	Improved Actuator Response; Reduced Backlash
Modified Motion Control Algorithm Architecture and Parameters	Improved Motion Fidelity; Eliminated Significant Artifacts

Light Weight Cockpit

The light weight cockpit (LWC) was designed to minimize the adverse effects associated with weight, inertia, and center of gravity (c.g.). Weight was a major design consideration throughout the construction of the LWC. The F-14 EFM Program cockpit weighed 2246 lbs. where as the LWC weighs only 1350 lbs. In addition, the center of gravity was placed closer to the gondola center of rotation. Reduced weight and inertia allow for increased g-onset and decreased phase lags in rate and acceleration response. Placing the c.g. closer to the center of gondola rotation acts to reduce the vertical force generated by the cockpit center of mass rotating around the gondola axis.

The combined reductions in weight, inertia, and c.g. position act to increase the natural frequency of the arm's vertical structural mode thereby reducing the potential for exciting the "hobby horse" mode.

II. DEVELOPMENTAL INVESTIGATION

Generic Aircraft Model Description

The aircraft model used to investigate motion fidelity characteristics of the DFS was designed to represent predicted future tactical aircraft capabilities. The aerodynamic database was developed from a collection of current technology military tactical aircraft [3]. The model assumed a rigid aircraft, with no corrections for aeroelastic effects at high Mach numbers. This generally resulted in significantly more control power available, particularly in the lateral axis.

The propulsion model was based on a modified General Electric F110 engine providing a maximum thrust to weight ratio of approximately 1:1 at sea level, static conditions. The flight control system was designed to provide handling qualities adequate for use during developmental investigations of the DFS's motion fidelity. Stability augmentation was provided in the pitch and yaw axis through rate feedback with proportional gains scheduled primarily on dynamic pressure, q. The lateral axis provided the pilot with roll rate command and feedback.

PST maneuvering was provided through direct deflection of the thrust plumb using longitudinal and lateral control inputs. Command inputs provide pitch and yaw moments based on a fixed moment arm multiplied by the magnitude of the thrust vector components deflected through vertical (τ) and horizontal (ε) angles (see figure 1). Moments generated by the thrust vectoring (TV) system were commanded by the pilot by depressing the paddle switch on the stick while commanding lateral and/or longitudinal stick. A block diagram of the TV control system architecture is provided in Figure 2. Note that TV can engage at any flight condition, however, the gains are based on dynamic pressure, q, with TV control power inversely proportional to q.

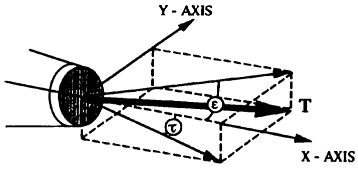
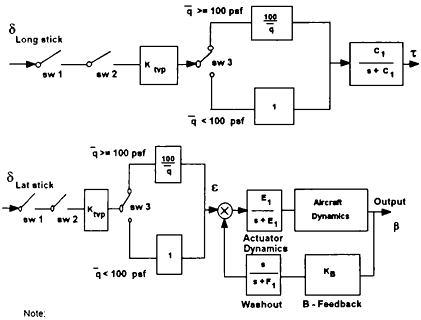


Figure 1: Thrust Vector Angles ε and τ



Note:
sw 1: Operator Station Enable
sw 2: Pilot Station Enable
sw 3: Dynamic Pressure Gain Compensation

Figure 2: Thrust Vector Control Law Architecture of The Generic Aircraft Model

The flight control system architecture and the aerodynamic database were specifically modified to provide adequate handling qualities for use in the light weight cockpit [4,5]. Departure resistance was provided by artificially setting static directional stability, $C_{n_\beta} \geq 0$ at all flight conditions. This resulted in the aircraft having essentially no nose slice or directional departure tendencies at or beyond stall AOA. Overall flying qualities were satisfactory for the investigation and development of motion fidelity characteristics.

Axes Systems for Aircraft and Centrifuge Simulations

The axes system used to express rates and accelerations in the DFS is similar to the conventional body-axis system familiar to pilots and aircraft engineers. Figure 3 shows the relationship of the DFS axis system to the conventional aircraft system. When the cockpit is installed facing tangentially, the A- and B-gimbal axes of the DFS coincide with the aircraft roll and pitch axis respectively. DFS arm rate, ω, is positive counter clockwise relative to the circular path of the gondola.

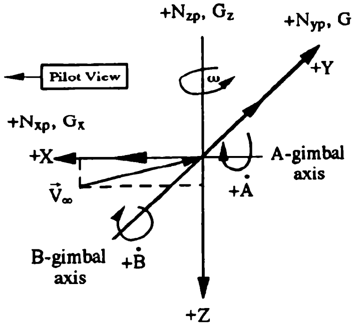


Figure 3: Relationship of DFS and Aircraft Axes Systems

All accelerations used to drive the CCA were referenced to the pilot's station. The generic fighter aircraft modeled the pilot's seat location coincident to the y- and z- body axis c.g. position (zero distance relative to the y and z aircraft c.g. locations), and at a realistic non-zero x distance ahead of the aircraft c.g. location.

Centrifuge Control Algorithm

The CCA is the key to good motion fidelity for centrifuge flight simulation. Without it the centrifuge recreates the aircraft G vector with very good accuracy but angular accelerations are created which produce unrealistic flight sensation. Consider the concept diagram in figure 4 as a representation of the processes occurring during pilot control of the aircraft. The pilot operates as a controller for the aircraft by deciding where he wants to go, moving his controls, sensing how both the controls and the aircraft are responding and adjusting the controls accordingly. (Clouds represent human sense and control functions.)

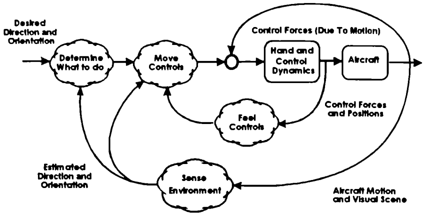

Desired Direction and Orientation
Determine What to do
Move Controls
Control Forces (Due To Motion)
Hand and Control Dynamics
Aircraft
Control Forces and Positions
Feel Controls
Estimated Direction and Orientation
Sense Environment
Aircraft Motion and Visual Scene

Figure 4 : Concept Diagram - Pilot Controlling the Aircraft

In a simulator the same process occurs with a compensation system controlling the simulator motion system replacing the aircraft. The desire is to equivalence the orientation response for these two systems. For the case of simulated motion a simple linear example conceptualizes the general process involved in the development of the CCA. Assuming an aircraft model (AM), a pilot sensory estimation model (PSEM), a motion system (MS), and a motion compensator (MC) the resulting orientation responses can be made equal.

$$AM \bullet PSEM = MC \bullet MS \bullet PSEM \qquad (1)$$

The construction of the motion compensator becomes as shown in figure 5.

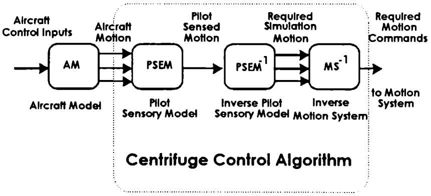


Figure 5: Simulation Motion Control Compensation

The centrifuge control algorithm (CCA) maps the simulated aircraft generated angular rates and linear accelerations into appropriate centrifuge gimbal angle and arm rate command responses. The CCA is designed to provide motion cues simulating an equivalent motion environment (as perceived by the pilot) representative of the current aircraft state. The architecture of the CCA is based on two components:

1. Vector transformations of aircraft body axes angular rates and linear accelerations to the centrifuge coordinate systems.

2. The Pilot Sensory Estimation Model (PSEM) due to Crosbie[6,7] developed with data from Cohen[8] estimates perceived orientation for both the aircraft pilot and the centrifuge pilot thus providing a mapping of the linear/rotational 6 degrees-of-freedom (DOF) aircraft motion environment to the purely rotation 3 DOF environment of the DFS. An example of the simplified roll axis PSEM is shown in figure 6.

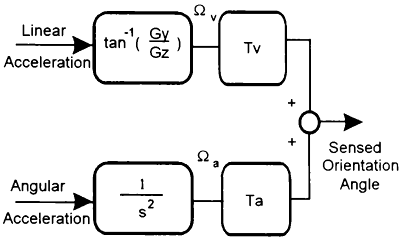


Figure 6: Simplified Roll Axis PSEM

Where response characteristics from Crosbie[7] are:

$$T_a = \frac{12.48\,s}{s^2 + 12.5\,s + 25} \qquad (2)$$

and:

$$T_v = \frac{6.66\,s + 5.0}{s^2 + 10.5\,s + 5.0} \qquad (3)$$

Equivalencing pilot sensed orientation in the aircraft and in the centrifuge gives the following relationship: (Superscripts denote either aircraft pilot (a), centrifuge pilot (c) or radial G (r).)

$$\Omega_v^a T_v + \Omega_a^a T_a = \Omega_v^c T_v + \Omega_a^c T_a \qquad (4)$$

The motion of the centrifuge roll axis primarily responds to recreate the aircraft G vector in the presence of radial acceleration. The roll axis constraint is:

$$\sin^{-1}(\frac{Gy}{Grv}) = A + \tan^{-1}(Gr) \qquad (5)$$

Approximating Grv with Gz, which is true for small angles, and identifying terms with those used in (4) gives:

$$\Omega_v^c \approx \Omega_a^c + \Omega_v^r \qquad (6)$$

Solving for the roll axis angular command angle shows how the characteristics of the sensory model filter could be directly incorporated into the CCA.

$$\Omega_a^c = (\Omega_v^a - \Omega_v^r)(\frac{Tv}{Ta + Tv}) + \Omega_a^a(\frac{Ta}{Ta + Tv}) \qquad (7)$$

The CCA as shown in figure 3 is completed by including the inverse motion system response characteristics.

This development was used to provide the form for the CCA. It was helpful as a model and discussion aid. The direct incorporation of these models has not yet been performed. The goal is to remove the "black art" from the control algorithm development changing the controls only as a function of the PSEM.

G-Bias Function

A G-bias function shown in figure 7 was used to allow a sense of acceleration unloading when the aircraft transitions to less than 1 G. It also reduces the amount of gimbal axis rotation needed for radial and tangential G coordination. It maps aircraft normal acceleration into centrifuge acceleration giving a 1:1 correspondence above 3 G.

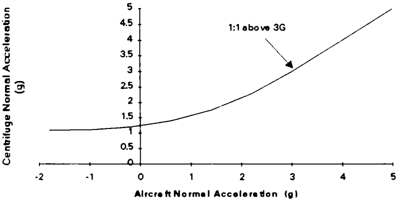


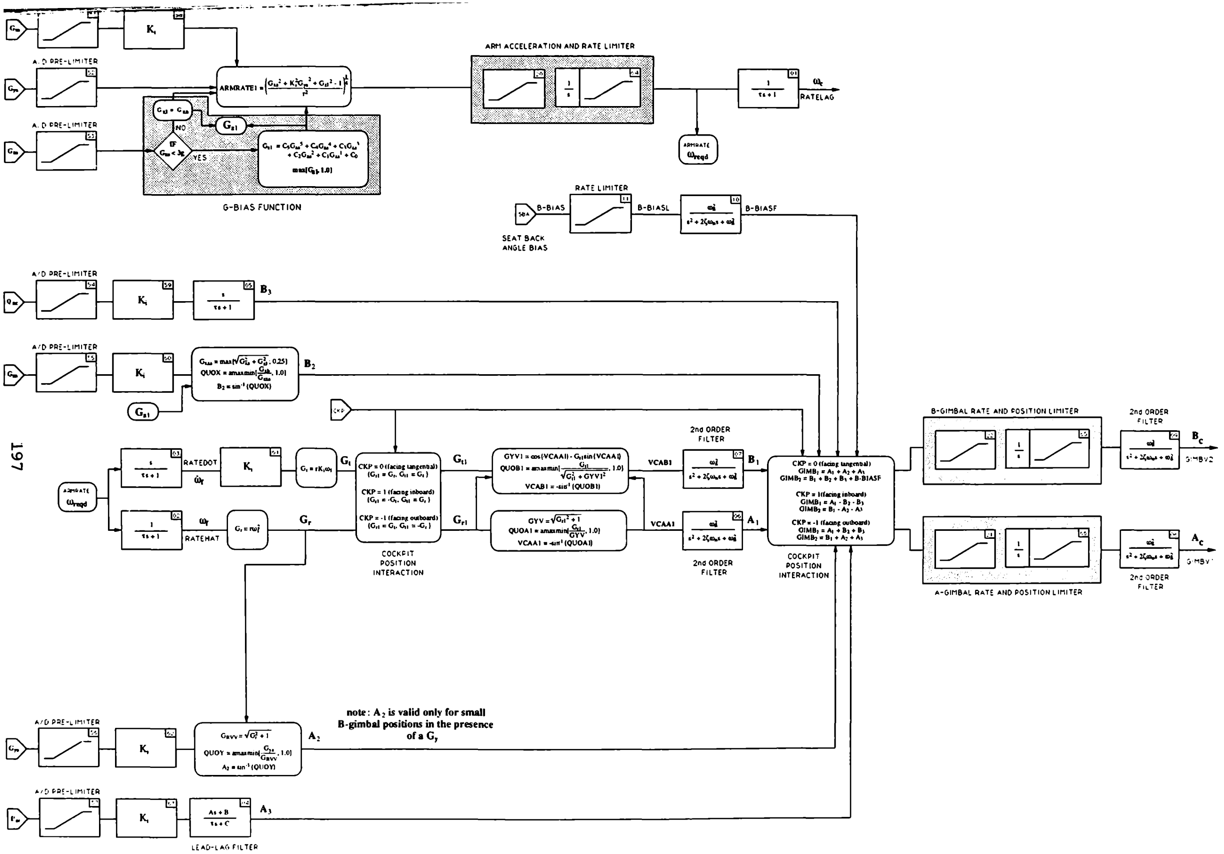
Figure 7: G-BIAS FUNCTION

The PSEM was experimentally designed to estimate a human perception of a given orientation from rates and accelerations of the motion environment acting on the human. The perceptual portion of the CCA was based on the PSEM with constraints imposed by the mechanical characteristics of the DFS (actuator dynamics, gimbal rate and position limits, structural considerations, etc.). The gains were designed to provide adequate thresholds in pilot response perception and omit excitation of unwanted response characteristics of the motion system. The current version of the CCA used during closed-loop simulation is presented in figure 8.

CCA Algorithm Improvements

Initial tests of the light weight cockpit uncovered a bank angle oscillation phenomena which had not been evident in previous experiments. A 1 to 2 Hz oscillation was excited with each lateral stick input, no matter how small the input. In addition, large rudder inputs (> 1/3 full throw) excited the roll oscillation. Data analysis showed that the oscillation was not being command by the CCA, but was most likely the result of a combination of drive torsion and backlash characteristics associated with the DFS gimbal axis drive system. Barring a re-design of the gimbal drive system, it was decided to attempt to minimize the unwanted mechanical characteristics through proper shaping of the servo command signals. The challenge was to prevent the effect of torsion and backlash from corrupting motion fidelity while keeping response onset and magnitudes at adequate thresholds to satisfy pilot perception requirements.

Tests were conducted to define the response characteristics of the gimbal servos. Sinusoidal inputs of various frequencies were used to drive the A- and B-gimbal drive axes while recording the input and output data. System identification analysis was performed to derive a linear transfer function. The original input data was then used with the new transfer function to estimate the actual response. The estimated output was compared with the actual output, and it was found that a pure delay, in combination with a first order transfer function, was needed to accurately model servo response characteristics. The final servo models are presented in figure 9. These results agreed very closely with the results of an investigation conducted in 1969 [9].


A. D PRE-LIMITER
A. D PRE-LIMITER
K
ARMRATE1 = $\left(\frac{G_{zz}^2 + K_i^2 G_{yz}^2 + G_{z1}^2 - 1}{r^2}\right)^{\frac{1}{4}}$
$G_{z1} = G_{zz}$
NO
IF $G_{zz} < 3g$
YES
G_{z1}
$G_{z1} = C_5 G_{zz}^5 + C_4 G_{zz}^4 + C_3 G_{zz}^3 + C_2 G_{zz}^2 + C_1 G_{zz}^1 + C_0$
max[G_{z1}, 1.0]
G-BIAS FUNCTION
ARM ACCELERATION AND RATE LIMITER
$\frac{1}{s}$
$\frac{1}{\tau s + 1}$
ω_r
RATELAG
ARMRATE
ω_{reqd}
RATE LIMITER
SBA
B-BIAS
B-BIASL
B-BIASF
SEAT BACK ANGLE BIAS
A/D PRE-LIMITER
K_i
$\frac{s}{\tau s + 1}$
B_3
A/D PRE-LIMITER
K_i
$G_{xzz} = \max(\sqrt{G_{xz}^2 + G_{z1}^2}; 0.25)$
QUOX = amaxmin[$\frac{G_{xz}}{G_{xzz}}$, 1.0]
$B_2 = \sin^{-1}$ (QUOX)
B_2
G_{z1}
CKP
$\frac{s}{\tau s + 1}$
RATEDOT
$\dot{\omega}_r$
K_i
$G_t = rK_i\omega_r$
G_t
ARMRATE
ω_{reqd}
$\frac{1}{\tau s + 1}$
ω_r
RATEHAT
$G_r = r\omega_r^2$
G_r
CKP = 0 (facing tangential) ($G_{r1} = G_r$, $G_{t1} = G_t$)
CKP = 1 (facing inboard) ($G_{r1} = -G_t$, $G_{t1} = G_r$)
CKP = -1 (facing outboard) ($G_{r1} = G_t$, $G_{t1} = -G_r$)
COCKPIT POSITION INTERACTION
G_{t1}
G_{r1}
GYV1 = cos(VCAA1) - G_{r1}sin(VCAA1)
QUOB1 = amaxmin[$\frac{G_{t1}}{\sqrt{G_{t1}^2 + GYV1^2}}$, 1.0]
VCAB1 = -sin^{-1} (QUOB1)
GYV = $\sqrt{G_{r1}^2 + 1}$
QUOA1 = amaxmin[$\frac{G_{r1}}{GYV}$, 1.0]
VCAA1 = -sin^{-1} (QUOA1)
VCAB1
VCAA1
2nd ORDER FILTER
$\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$
B_1
A_1
2nd ORDER FILTER
CKP = 0 (facing tangential)
$GIMB_1 = A_1 + A_2 + A_3$
$GIMB_2 = B_1 + B_2 + B_3$ + B-BIASF
CKP = 1(facing inboard)
$GIMB_1 = A_1 - B_2 - B_3$
$GIMB_2 = B_1 - A_2 - A_3$
CKP = -1 (facing outboard)
$GIMB_1 = A_1 + B_2 + B_3$
$GIMB_2 = B_1 + A_2 + A_3$
COCKPIT POSITION INTERACTION
B-GIMBAL RATE AND POSITION LIMITER
A-GIMBAL RATE AND POSITION LIMITER
$\frac{1}{s}$
2nd ORDER FILTER
B_c
GIMBV2
A_c
GIMBV1
2nd ORDER FILTER
A/D PRE-LIMITER
K_i
$G_{RVV} = \sqrt{G_r^2 + 1}$
QUOY = amaxmin[$\frac{G_{ya}}{G_{RVV}}$, 1.0]
$A_2 = \sin^{-1}$ (QUOY)
A_2
A/D PRE-LIMITER
K_i
$\frac{As + B}{\tau s + C}$
A_3
LEAD-LAG FILTER

note : A_2 is valid only for small B-gimbal positions in the presence of a G_y

Figure 8: Centrifuge Control Algorithm For Closed-Loop Flight Simulation

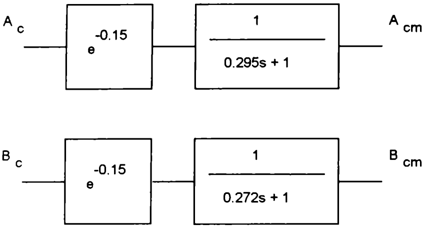


Figure 9: Experimentally Derived A- and B- Gimbal Axis Servo Models

A batch simulation model of the CCA's roll command structure was developed and analyzed specifically to omit the roll oscillation problem. Assuming that roll response was represented by an equivalent first order system (classical aircraft response), the response to the washout filter was causing reverse A-gimbal command signals upon neutralization of the pilot input (see figure 10).

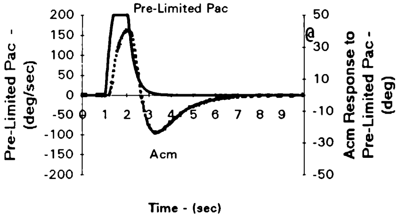


Figure 10: Original A-Gimbal Command Response to Step Input Assuming First Order Aircraft Dynamics in the Roll Axis

These reverse command inputs were perceived by pilots as unrealistic. It was perceived as an initial delay followed by a rapid onset as if the gondola were attempting to catch up with the aircraft command. In any case, it was theorized that the sudden reversal in servo command signal was exciting the adverse mechanical characteristics associated with shaft torsion, gear freeplay, backlash, and inertia properties.

Using the batch model, it was shown that the reverse signal could be removed using a modified lead-lag in place of the original washout, and placing a second order filter prior to the actuator command signal. The batch model was used to vary the parameter values of each filter to force a desired response to the roll input. Several candidates were chosen and tested in the simulator. Initial tests were with gimbals only, followed by fully centrifuge simulation. The final selected parameters were:

$$\frac{1.8s}{4s+1} \qquad \text{(modified lead lag)} \qquad (8)$$

$$\frac{25}{s^2+10s+25} \qquad \text{(second order filter)} \qquad (9)$$

Manned dynamic testing proved the modified filters were successful in reducing the backlash without significantly altering the response onset characteristics of the original washout. Figure 11 shows the response of the new gimbal output to the pilot roll command.

To prevent possible pitch axis backlash caused by abrupt, large pitch rate commands, the second order filte was also placed in the B-gimbal command axis just prior to the actuator command block (see figure 8, CCA Algorithm). Dynamic testing showed that this filter was successful in preventing backlash with minimal effect on pitch response characteristics.

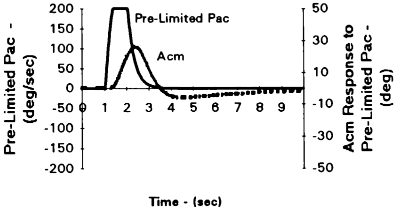


Figure 11: Modified A-Gimbal Command Response to Step Input Assuming First Order Aircraft Dynamics in th Roll Axis

III. RESULTS AND DISCUSSION

General

Motion fidelity of the DFS was investigated by engineers and tactical military pilots using the generic fighter aircraft simulation model. Both static (motion off, fixed base) and dynamic modes (motion on, centrifuge operating) were flown. After each pilot was familiarized with the flight characteristics of the generic aircraft model, they conducted a series of maneuvers designed to investigate motion fidelity characteristics. Throughout the evaluation, pilots were continuously prompted to provide comments and recommendations regarding motion fidelity.

Piloted investigations were conducted between 200 and 500 KCAS, sea level to 40000 ft. Axial acceleration (G_x) characteristics were investigated with airspeed changes imparted during commanded thrust and attitude changes. Normal acceleration characteristics (G_z) of the DFS were investigated during maneuvering flight tasks including bank-to-bank rolls, heading capture tasks, wind-up turns, and pure longitudinal stick inputs. Yaw rate and side force cues (G_y) were evaluated using rudder doublets and step roll control inputs. Pitch rate and acceleration cues were investigated with smooth and abrupt pitch command inputs including stick doublets. Bank angle, roll rate and roll acceleration characteristics were qualitatively investigated during normal operational flight tasks including 1-g 360 degree rolls, bank-to-bank rolls, and level turns.

Qualitative and quantitative data were documented throughout the evaluation. A post-flight brief was conducted to clarify information as necessary, and each pilot provided a written flight report. Overall, several key improvements have been documented. Noted improvements included better lateral acceleration cues, roll response cues, and improved rapid maneuvering characteristics without excitation of unwanted structural modes.

Gz Onset Performance

Gz onset performance characteristics are a function of the aircraft model as well as the DFS motion system. For this investigation, Gz/sec data were obtained by using DFS time history data corresponding with aircraft response to abrupt inputs. Gz/sec was computed from the average Gz felt by the pilot over the range of G commanded up to the manually set G limit of the DFS. Figure 12 outlines a typical response of the DFS to abrupt pitch inputs. Note that for this trace, a 3-g DFS limit was placed on the pilot. In this particular case, Gz/sec was 1.5 g/sec. Throughout the investigation, both quantitatively and qualitatively, the DFS was able to provide perceptually realistic Gz onset capability throughout the operational flight envelope of the generic fighter aircraft.

Translational Acceleration Response Fidelity

Translational acceleration (Gx, Gy, Gz) response fidelity was investigated using moderate and abrupt control inputs in all axes. Initial response fidelity addresses any response threshold, phase lag, and/or time delay between command input and DFS motion response. Pitch, roll, and yaw step inputs were conducted to evaluate onset characteristics in the respective aircraft x-, y-, and z-axis. In general, pilot comments indicated that there were no perceptible lags or delays in Gx and Gz. In addition, Gx and Gz thresholds were adequate in providing pilots with perceptually realistic response cues throughout the flight envelope. However, the ability of pilots to perceive Gy varied from pilot to pilot, and as a function of maneuver. During maneuvers for which pilots commented perceiving Gy cues, no response lags or delays were apparent.

Axial acceleration (G_x) characteristics were investigated during 1-g acceleration and deceleration tasks using throttle inputs, and during zoom climb and diving pullout maneuvers. Changes in axial acceleration due to small and moderate throttle movements provided realistic Gx response and magnitude cues with no discernible phase lag or time delay. Axial acceleration changes during zoom climbs and diving pullouts were also acceptable. However, large changes in Gx caused by large throttle movements resulted in excessive pitch up (acceleration) and pitch down (deceleration) sensations unrepresentative of the aircraft's state. Figure 13 provides a clear example of how large throttle movements resulted in excessive B-gimbal command signal, which were detected by the pilot as a pitch sensation as opposed to Gx. During developmental testing, it was shown that this artifact could be alleviated by reducing the CCA gain Kx. However, this approach runs the risk of removing axial acceleration sensations during other tasks involving Gx. Additional proposed corrections are currently being investigated.

Angular Rate Response Fidelity

Angular rate phase lag and delay characteristics were evaluated throughout the motion fidelity investigation. Phasing cues were analyzed by the pilot through comparison of initiation of command input with visual display update and perceived motion sensation. Analyses included moderate and abrupt control inputs in all axes. In general, pilot comments indicated that response onset was predictable with no apparent phase lags or delays. Pilot comments indicated that pitch up sensations were realistic; however, during abrupt pitchovers resulting in large negative Gz, pilots indicated that the negative Gz was perceived more as negative Gx. Figure 14 shows a time history of a pushover to 30 deg pitch attitude.

A typical 360 deg roll is presented in Figure 15. During rapid rolling maneuvers, pilots noted that roll rate and roll acceleration motion cues were perceptible. Lateral acceleration, Gy, cues were also evident during abrupt rolling maneuvers, but were not perceptible by all pilots.

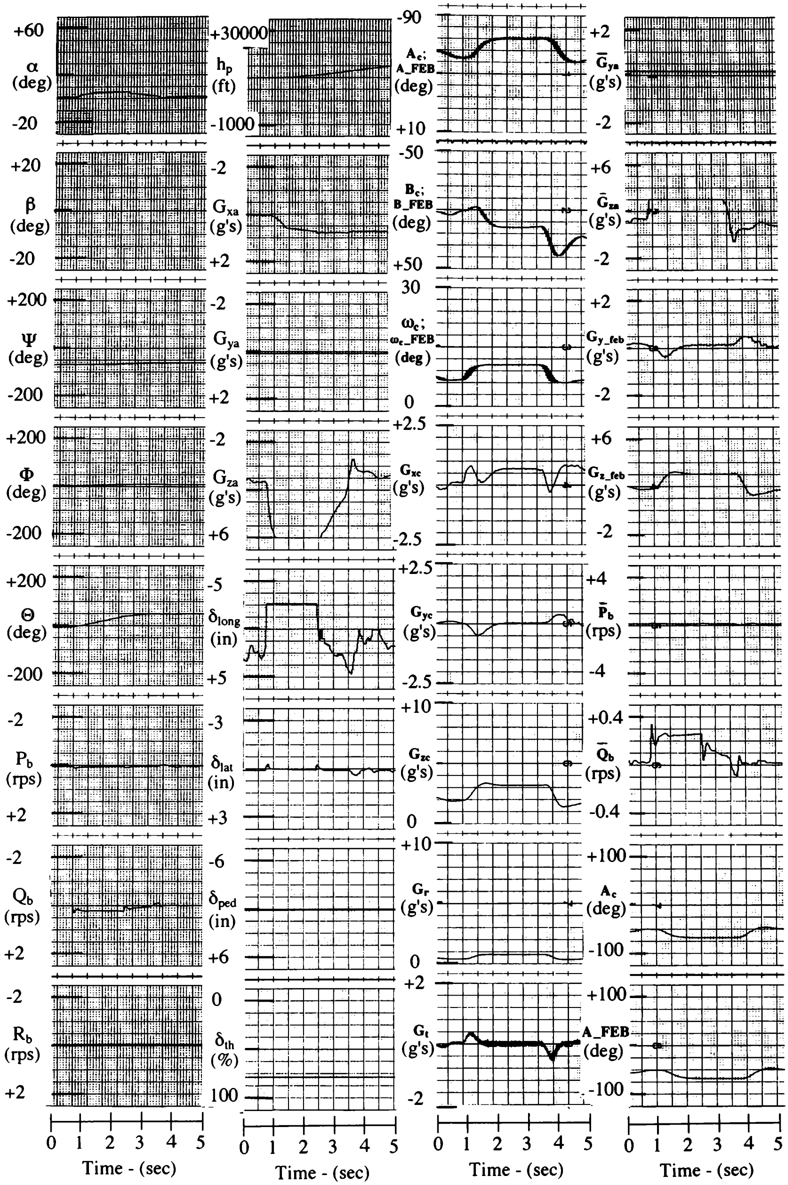


Figure 12: Typical Gz Onset Performance

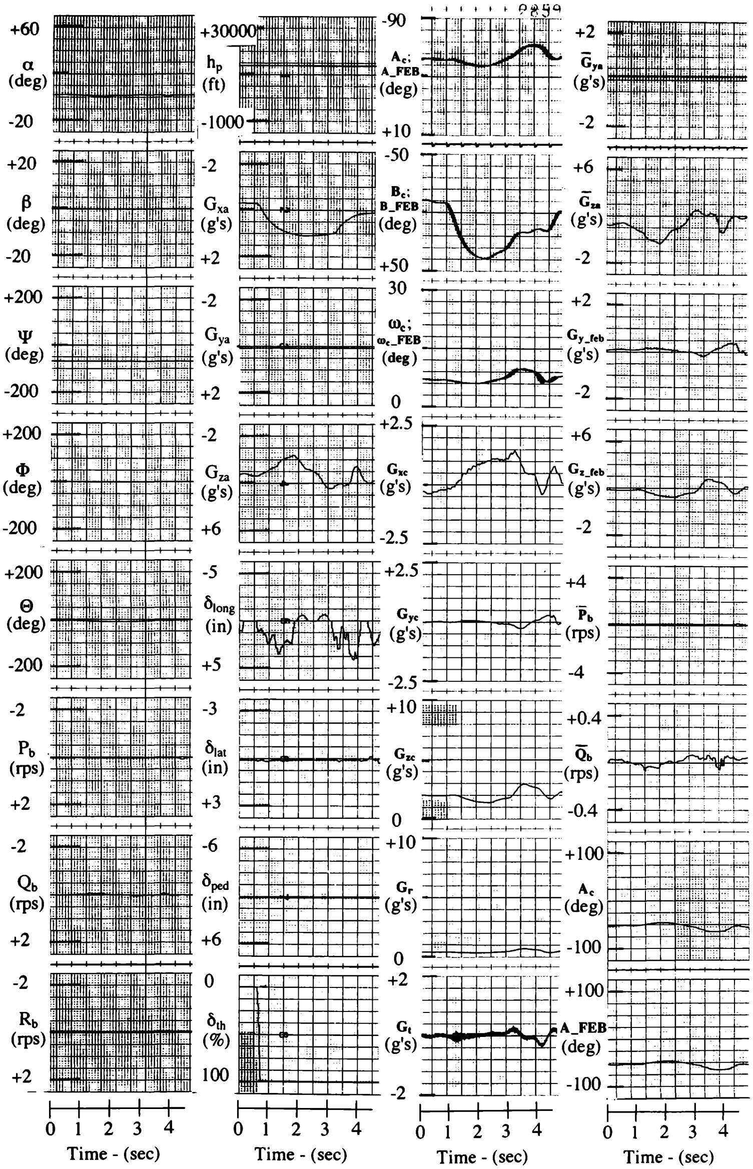


Figure 13: Rapid Change in Gx Using The Throttle

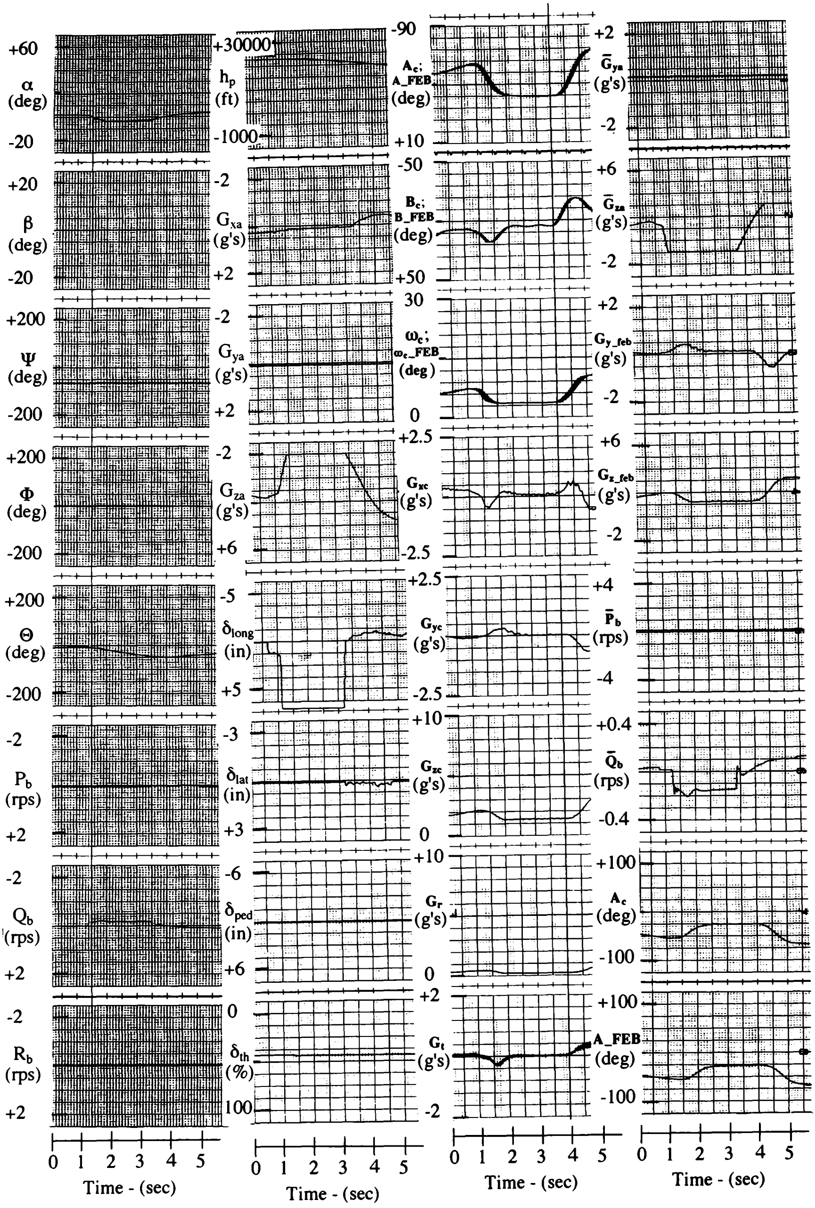


Figure 14: Push-Over To -30 Deg Pitch Attitude

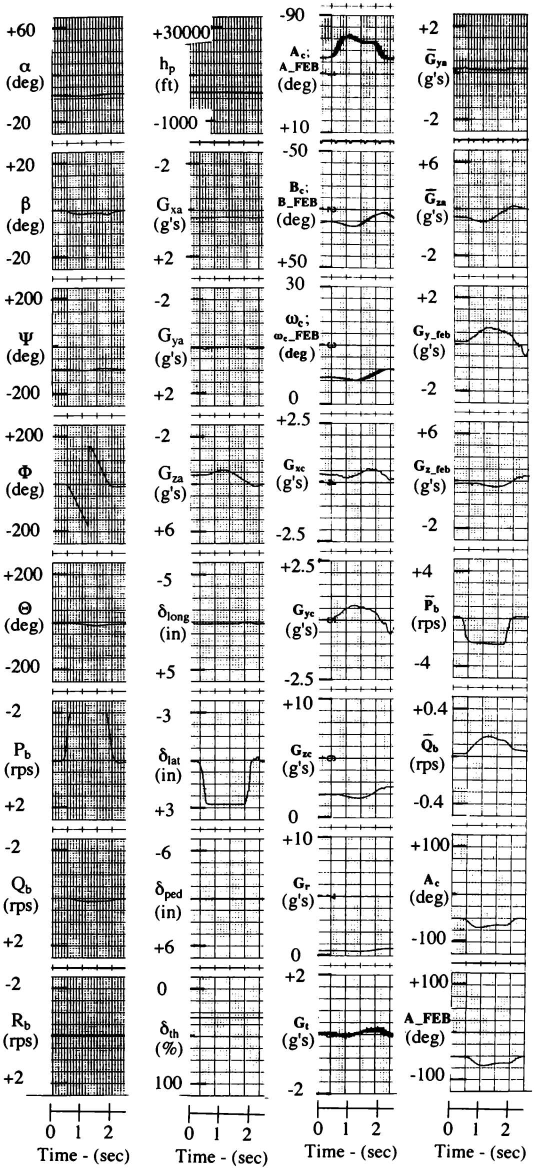


Figure 15: Abrupt 360 Deg Aileron Roll

Furthermore, most pilots stated that roll rate and lateral acceleration motion cues were difficult to perceive during slow rolling maneuvers. Currently, modifications to the CCA architecture are being considered to address how to increase roll axis thresholds without exciting unwanted artifacts and structural modes.

During high angle of attack rolling maneuvers with TV on (where rolling can result in considerable body axis yaw rate) lateral cues were low to negligible. This characteristic was not a surprise considering mild rolling maneuvers at normal AOAs and at HAOAs produce similar magnitude aircraft roll rates, and the previous condition had already indicated a need for increased perception thresholds. Figures 16 and 17 show time history traces of a moderate rate 360 deg roll at low AOA and a 360 deg roll initiated at HAOA with TV engaged, respectively.

Overall, the DFS has demonstrated improvements in rate response and onset fidelity using the newly modified CCA. The aforementioned deficiencies are currently being addressed. Success of the already implemented DFS modifications advocate a high probability of success in maturing the CCA to provide excellent motion fidelity throughout the flight envelope.

Disorientation

During DFS operations, certain unrepresentative sensations may be experienced by pilots. The principle contributions to this disorientation in a centrifuge are due to three factors. The Coriolus effect, the tangential G effect and a G gradient. The relative magnitudes with respect to arm radius of the physical stimulus generating these effects is shown in figure 18. Please note this figure is not a comparison of the effects of disorientation types. Head motions at low (1 to 3g) levels may result in some disorientation characterized by a sensation of rolling or pitching. This unrepresentative sensation, defined as the Coriolus effect, is due to the rotational environment of the centrifuge. The effect is not experienced in aircraft primarily due to the relatively large instantaneous turn radius of current designs. Although some increase in arm radius will minimize the Coriolus artifact, a large arm radius can result in tangential acceleration artifacts. Recall that a centrifuge designed to provide Gz through the normal component of acceleration associated with rotation (A_n) must first produce a tangential velocity, V, capable of sustaining it ($A_n = V^2/r$). While simulating rapid aircraft Gz onset, large, unrepresentative, tangential accelerations, A_t, will result. If A_t is used to supplement pilot perceived Gz, then unrepresentative rotations will be necessary to align the acceleration vector with the gondola. These rotations will very likely result in significant rate artifacts.

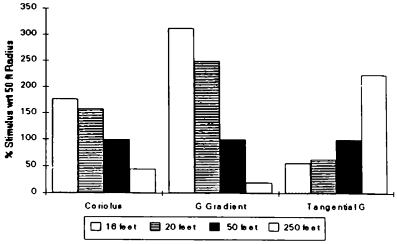


Figure 18: Effects of Radius on Disorientation Stimulus (Normalized by Category)

The reader should keep in mind that Coriolus effect does occur during maneuvering flight, but large aircraft turning radii typical of today's designs render it difficult to perceive. However, it is unknown how much Coriolus effect pilots will experience during PST maneuvering where the instantaneous turn radius may be as low as 200 ft.

During this evaluation, pilot comments indicated that strong Coriolus effect was perceptible near the plateau g of 1.55. However, pilot comments also indicated that the Coriolus effect tended to be less noticeable as g increased. In addition, each pilot's simulation session time tolerance increased as he/she accumulated more DFS flight (exposure) time. This observation supported the test results obtained during the F-14 HAOA and spin evaluations using the DFS [10].

In summary, the Coriolus artifact will be most evident at the plateau g. However, test results indicate that the Coriolus effect can be overcome with increased plateau g, exposure time, and pilot awareness of the artifact. This Coriolus effect is considered a nuisance trait versus an actual limitation of the DFS. It will only result in minimal limitations during the majority of flight tasks.

Advantages of Centrifuge Flight Simulation

With the developmental and procurement costs of aircraft rising at unprecedented rates, simulation is being increasingly relied upon throughout the design process. Fixed based and limited motion simulation has played a key part in successfully developing and assessing open-loop flying qualities and flight control system characteristics prior to entering the flight test program. However, due to the lack of important motion cues which close the feedback loop between the pilot and the

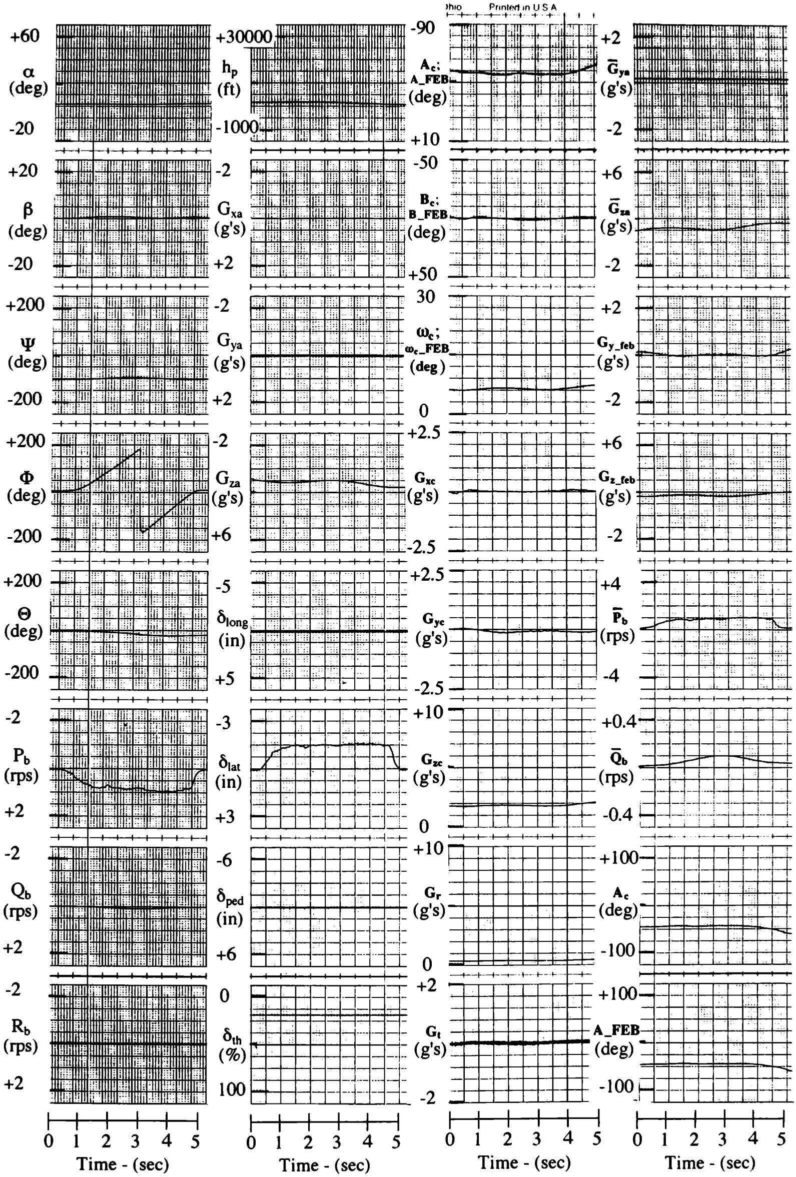


Figure 16: Moderate Rate 360 Deg Roll At Low AOA

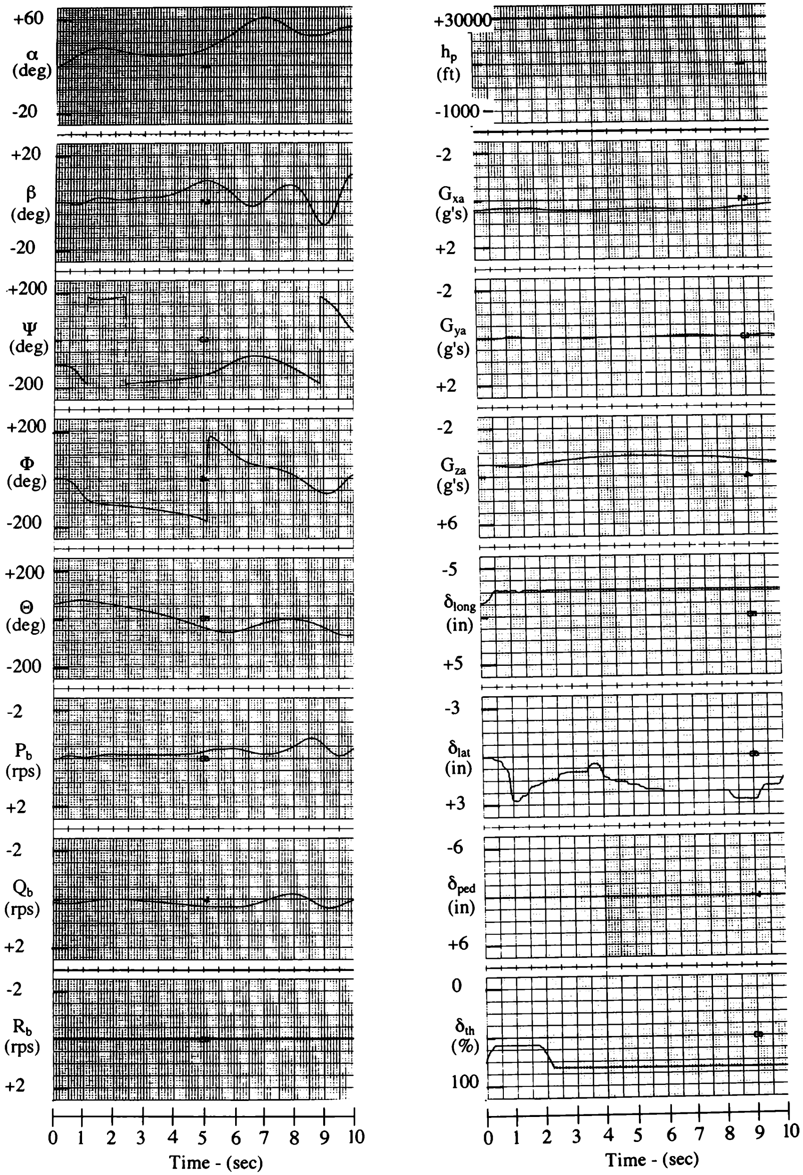


Figure 17a: 360 Deg Roll At HAOA (TV On)

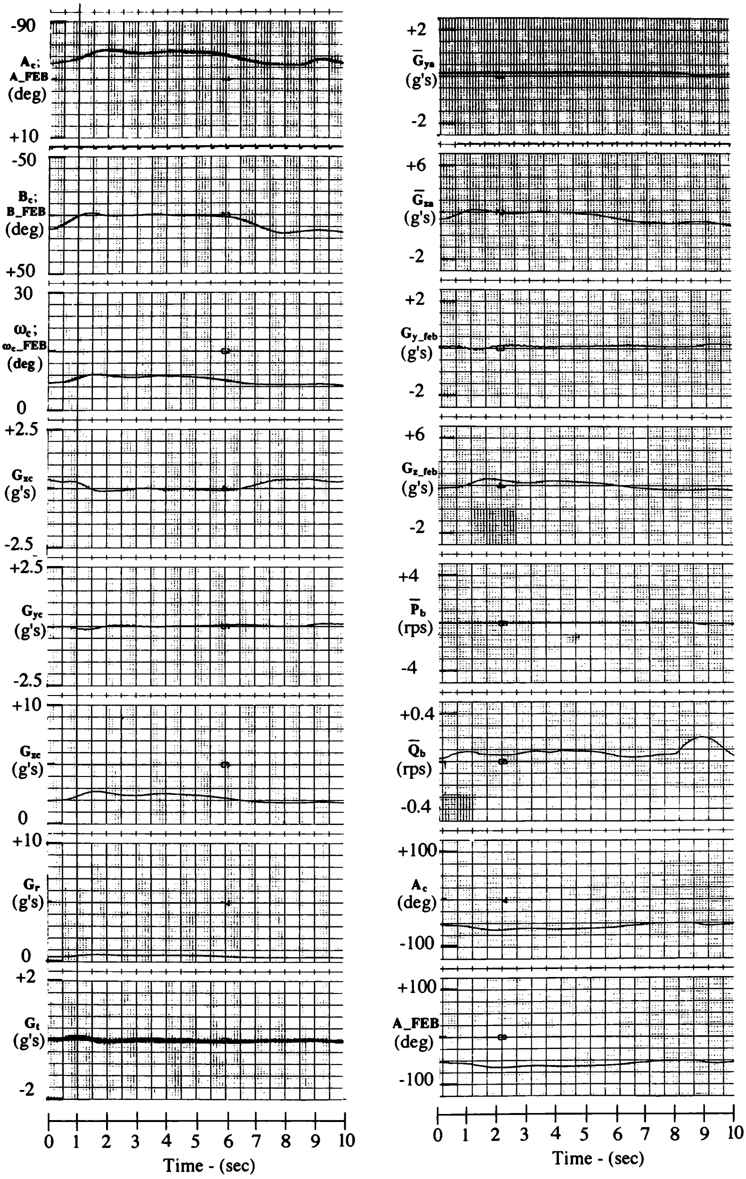


Figure 17b: 360 Deg Roll At HAOA (TV On)
(cont.)

aircraft, several undesirable closed-loop flying qualities characteristics go undetected prior to flight testing. Centrıfuge flight simulation offers a cost effective intermediary between fixed/limited motion simulation and flight test. The following paragraphs provide two cases of how it improved flying qualities assessments of the generic fighter aircraft model and flight simulation fidelity.

During static testing of the generic fighter aircraft model, roll sensitivity and control forces (roll rate per lb. of stick force input) were considered adequate by the military evaluation pilots. However, during dynamic evaluations, the same roll sensitivity and control forces were determined to be too high. Several reports document the difficulty in adequately assessing control sensitivities such as stick force per g, angular rate per controller force/displacement, and roll mode time constant in a reduced cue environment. The advantage of sustained motion simulation and the utility of the DFS in assessing closed loop flying qualities characteristics of aircraft is evident. As illustrated by this example, centrifuge simulation is clearly able to provide valuable insight for handling qualities improvements to aircraft designs by providing the sustained motion environment not available in conventional fixed and motion based simulators.

During PST maneuvering, the evaluation pilot departed the generic fighter following multiple aileron rolls at approximately 30 deg AOA. Prior to the maneuver the head-up display (HUD) ladder and angle of bank indicators had failed. Although visual cues were impaired by the aircraft's excessively oscillatory pitch and roll attitudes (see figure 19), minimum terrain cues in the visual scene database, and the limited 30 deg vertical field-of-view image system, the pilot was able to successfully recover the aircraft. The sustained motion environment provided adequate perceptual motion cues allowing the pilot to recover the aircraft. In fact, physiology data showed the pilot's heart rate increased from an average 75 bpm during normal flight tasks, to over 95 immediately following departure. This indicated the magnitude to which the DFS was able to convince a pilot that he was actually operating an aircraft.

Evaluation of CFS as a Technology

Several criticisms of CFS technology quality and practicality must be addressed directly.

a. The centrifuge angular motion produces a disorienting sensation referred to as the "Coriolus effect". This is a characteristic of perceived roll and pitch motion associated with head movement which can lead to motion sickness. It is most noticeable at the operating plateau level (aircraft 1-g trim), and its effect diminishes significantly with increasing g. It is controlled by limiting head motion in the test design until a tolerance develops.

b. The centrifuge motion can cause motion/simulator sickness. In practice this effect is comparable to other simulation methods, as well as flight itself. There is a tendency toward motion sickness typically after a half hour of intensive aircraft maneuvering. Several techniques are used to reduce it. Among them are: 1) using the perception based motion control algorithm; 2) developing a tolerance to the motion usually after two or three exposures; 3) using appropriate test design which limits pilot head movement and terminates test runs before sensations become acute; and 4) maintaining a good correspondence between the visual display system and the expected aircraft motion. Also, the man-in-the-loop nature of CFS reduces these effects compared to pure man-out-of-the-loop centrifuge exposures used for physiological studies.

c. CFS does not exactly replicate aircraft linear and angular accelerations. Steady state linear accelerations are matched according to a defined schedule, which is 1-to-1 above 3G. Angular accelerations are replicated only within the constraint of maintaining proper orientation sensation . The fact that CFS can come close to achieving actual aircraft accelerations evokes this criticism not considered for other ground based simulation.

d. CFS technology is practical. The two main cost drivers are radius and visual display system. CFS ha been demonstrated in devices with a main axis rotation radius as short as 8 feet. Higher quality implementations such as the DFS increase the range of applicable tasks.

e. Valid application of simulation testing to aircraft performance is task dependent. The fact that CFS cai recreate a subset of the aircraft motion environment not otherwise obtainable implies that there is a class of tasks for which DFS provides better result transfer than other simulation methods (see table II).

f. Weight and volume constraints combined with the need for equipment ruggedization are limitations of the technology. A manifestation of this is a limited field of view of the out-the-window display system when compared to a dome simulator.

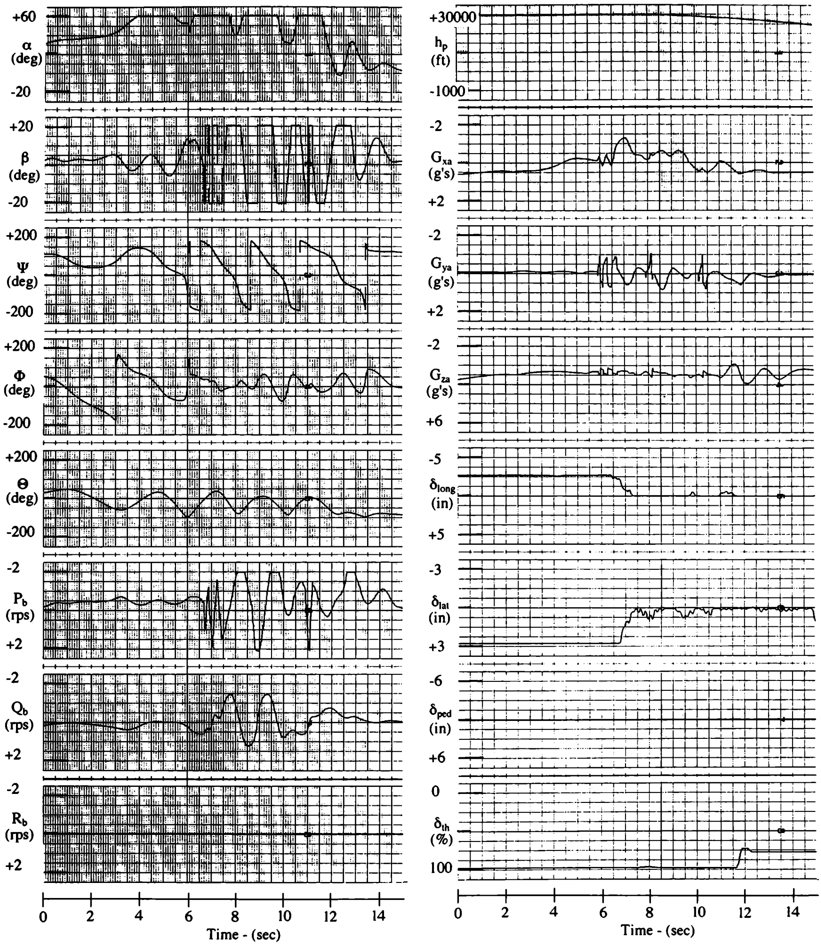


Figure 19a: Departure Of The Generic Fighter Aircraft

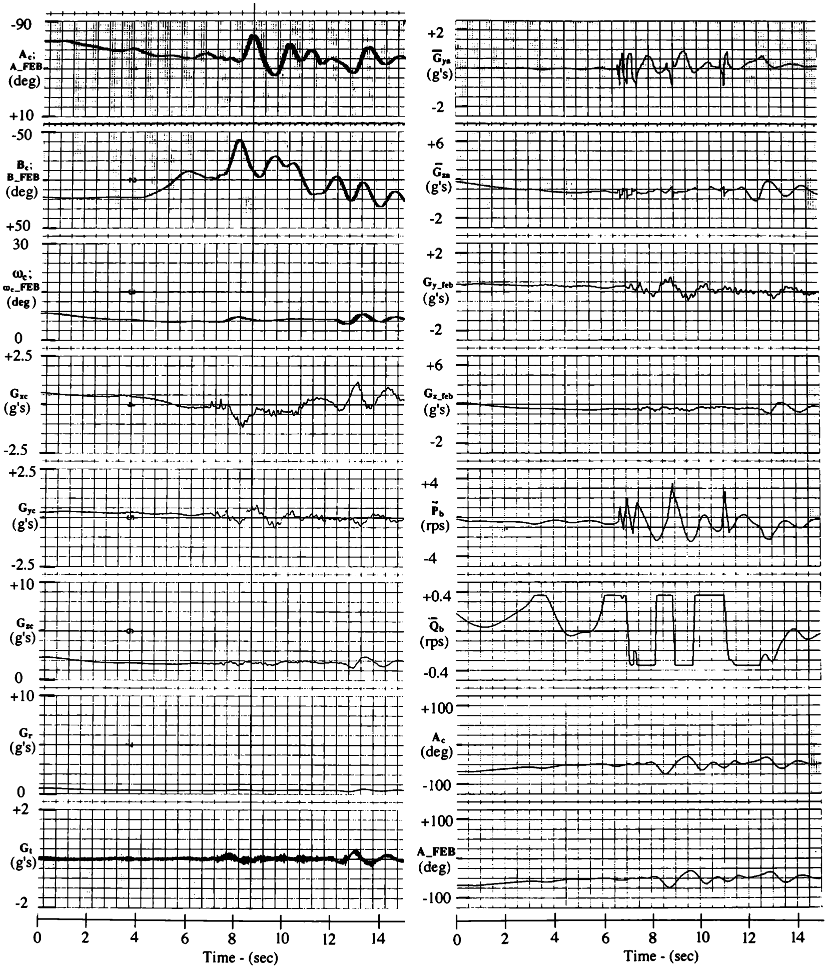


Figure 19b: Departure Of The Generic
Fighter Aircraft (cont.)

Table II: Motion Characteristics for Different Ground based Simulation Types

Motion Characteristic	Aircraft (Existing & Potential)	Static Flight Simulation	Limited Motion Flight Simulation	Centrifuge Flight Simulation
Gx	±1.5g	0	direction & transient	-10, +15
Gy	±1.5g	0	direction & transient	-10, +10
Gz	-4, +9g	1	direction & transient	+1, +15
p (°/sec)	± 400	0	transient	transient
q (°/sec)	± 100	0	transient	transient
r (°/sec)	± 200	0	transient	transient
Disorientation	Yes	Yes	Yes	Yes
Sickness	Yes	Yes	Yes	Yes

IV. CONCLUSIONS

This paper outlines the developmental efforts of the Dynamic Flight Simulator since 1986. A generic fighter aircraft model with capabilities representative of future fighter aircraft (including TV and EM) was developed and hosted on the DFS. The current DFS configuration and associated motion fidelity characteristics were evaluated and documented. The CCA has been successfully modified to eliminate essentially all undesirable mechanical characteristics with no apparent effect on motion fidelity.

Overall, the results of developmental evaluations to date indicate that the DFS is capable of producing a pilot perceived motion environment consistent with that experienced during actual flight. Several improvements have been noted over previous configurations. Specifically, the lateral response characteristics of the DFS have improved, and the potential to develop Gy sensations was demonstrated. It is highly probable that the DFS will be capable of simulating lateral-directional motions of high performance aircraft up through HAOA flight conditions with acceptable fidelity to perform limited flying qualities and human factors investigations. The DFS exhibits strong potential to be a cost effective bridge between current simulators and flight.

The principal advantages of the DFS will occur for RDT&E experiments during aggressive closed-loop piloting task where flying qualities and human factors issues are being investigated. Such issues include:

a. Display Design (on and off axis)

b. Situational Awareness

c. Spatial Disorientation

d. Air Combat Maneuvering

e. Air-to-Ground Maneuvering

Perhaps the most current issue today directly relates to the flight environment encountered during HAOA maneuvering. Pilots from both the X-29 and F-18/HARV programs who have maneuvered at HAOA underscored the need for improved displays [2]. In addition, the key challenges to future HAOA flight developments will include the operational considerations of situational awareness, energy management, and weapons release. Centrifuge flight simulation exhibits a high potential to provide a low cost approach to investigating these issues from design to operational readiness.

V. FUTURE EFFORTS

Future developmental efforts will be aimed at conducting limited flying qualities and human factors experiments which utilize the current and projected near-future capabilities of the DFS. To best support this effort, modifications to the generic fighter aircraft model will be incorporated to ensure proper handling qualities and provide improved TV and trim capability through 90 deg AOA. Also, the CCA will be slightly modified to reduce the excessive pitch sensation associated large magnitude Gx, and improve motion fidelity during pushovers and low rate rolling maneuvers.

Improvement to the visual system are also planned. The database will be upgraded to provide added depth perception and height cues via improved terrain resolution (trees, houses, roads, hills, etc.). Furthermore, efforts will be made to increase the vertical and horizontal field of view of the visual system for enhanced air combat maneuvering (ACM) capability.

Perhaps the most important future endeavor will be the development and validation of a batch simulation model of the DFS. This model will account for mechanical characteristics and thereby allow for off-line gain optimization to ensure best motion fidelity. The model will be validated using actual DFS data, and may require parameter identification to extract mechanical characteristics which are difficult to measure. This model will provide a low cost approach to refining the control algorithm off-line and maximize the DFS's potential for an expanded role as a high fidelity, man-in-the-loop simulator.

Finally, with regard to hardware, future efforts will concentrate on building a lightweight center-stick cockpit and improving the digital data collection/manipulation capability of the facility.

REFERENCES

1. High-Angle-of-Attack Technology Conference, NASA Langley Research Center, Oct 91.

2. High-Angle-of-Attack Projects and Technology Conference, NASA Conference Publication 3137, vol. 1, Apr 92.

3. Anonymous, "Generic Fighter Aircraft FASTER Simulation Model Report - Final Report", Veda Report 33159-91U/P3351-005, Jan 91.

4. Calvert, J. F., "Preliminary Evaluation of the Dynamic Flight Simulation Motion Fidelity for Manned Flight Simulation in a Motion Environment", NAWC-AD-WAR Working Report, in preparation.

5. Anonymous, "Software Product Specification for the DFS Aircraft Thrust Vector Experiment", Veda Report 33544-92U/P3553-011, Mar 92.

6. Crosbie, R.J., "Application of Experimentally Derived Pilot Perceptual Angular Response Transfer Functions", AIAA-83-1100-CP, AIAA Flight Simulation Technologies Conference and Technical Display, Jun 83.

7. Crosbie, R.J. and Kiefer, D.A., "Controlling the Human Centrifuge as a Force and Motion Platform for the Dynamic Flight Simulator", AIAA-85-1742, AIAA Flight Simulation Technologies Conference, Jul 85.

8. Cohen, M.M., "Human Performance During Exposure to Combined Linear and Angular Accelerations", NASA Contract MPRT-8477C.

9. Fortenbaugh, R. L., "Assessment and Control of the Undesired Components of Human Centrifuge Acceleration Response to a Single Axis Command", NADC-AM-6923, Jun 69.

10. Bischoff, D. E., Poole, D. A., and Eyth, J., Jr., "Flight Simulation Fidelity in a Total G-Force Environment", AIAA Paper 85-1741, AIAA Flight Simulation Technologies Conference, Jul 85.

AIAA-92-4158-CP

ALGORITHMIC IMPROVEMENTS FOR SIMULATOR MOTION DRIVE

Soren LaForce
Senior Systems Engineer
SYRE, Inc., NASA-Ames Research Center, Moffett Field, CA

and

John W. Bunnell
Manager, Simulation Programs Department
SYRE, Inc., NASA-Ames Research Center, Moffett Field, CA

Abstract

Contemporary simulator motion drive algorithms typically are designed in an analog (continuous) environment, but are implemented in a digital (discrete) environment. The intended continuous system, specified as frequency domain (Laplace transform) transfer functions, may not be represented properly by the algorithms used for digital implementation. The motion drive software in use with the Vertical Motion Simulator at Ames Research Center was investigated recently; the original algorithms (Euler) were changed to a state transition method. Comparison of the frequency responses of the original and new implementations showed that the state transition method more closely approximates the desired analog responses. In addition, test pilots who evaluated both implementations preferred the motions generated with the state transition method over those generated with the Euler integration method.

Nomenclature

S	Laplace operator	- sec^{-1}
T	Cycle Time	- sec
t	Time	- sec
ϕ	Roll angle	- radians
ω_p	Roll Axis Corner Frequency	- sec^{-1}
ζ_p	Roll Axis Damping Ratio	- Nondimensional
Gx	Longitudinal Axis High Frequency Gain	- Nondimensional
Gp	Roll Axis High Frequency Gain	- Nondimensional
Ka	Acceleration Feedforward Gain	- sec^2
Kv	Velocity Feedforward Gain	- sec

Introduction

In general, motion simulators are used to provide vestibular cues to pilots sufficient to improve simulation realism compared to that attainable in a fixed base (no motion) environment. They usually do not attempt to provide the same accelerations that would be experienced by the pilot of the simulated aircraft. With the exception of airborne simulators (variable stability aircraft), attempting to provide the same accelerations calculated in the aircraft math model would require a motion base with sufficient travel to be economically, if not technically, infeasible.

Most motion systems used in flight simulation provide high frequency vestibular cues by the application of high-pass filters to the accelerations calculated to occur at the pilot's location. The filtered high frequency accelerations may then be integrated to provide rate and position information that is used to drive the motion base. Low frequency accelerations (defined as those attenuated by the high-pass filters) may also be provided for the lateral and longitudinal axes by tilting the cab to introduce a gravitational reaction into the desired axis. Low frequency cues in the vertical and rotational axes are not normally provided.

Research in human perception indicates that translational accelerations (specific force) and angular velocities are the primary vestibular stimuli [1]. In addition, although specific force may be perceived down to very low frequencies, there is a lower limit to angular velocity perception at about 0.05 radian/second [1]. This perceptual attenuation suggests that the lack of low frequency rotational rate and acceleration cues may not adversely affect the simulation fidelity. The basis for this type of motion washout, both theoretical and empirical, is discussed in references 1, 2, 3, and 4.

Figure 1 outlines the system used in the Vertical Motion Simulator (VMS) facility to provide motion cues to the pilot. In addition to the translational and rotational cues already discussed (TRANSL and ROTATE in Figure 1), there are two translational paths used by the VMS that may not be used in other facilities. The first (TRNCRD), exploits the large lateral travel (plus or minus 15 feet of usable travel) to maintain coordinated turns without the use of excessive rotational washout. The second (INDUCE), generates a translational command which compensates for the fixed rotation center of the VMS, which is several feet below the pilot's location. The primary path in the motion washout system used in the VMS facility is a second-order high-pass filter. The input to this filter is the pilot station acceleration as calculated by the aircraft model and system kinematic equations. The output, the desired simulator acceleration, is then integrated as required to provide the desired simulator velocity and position commands.

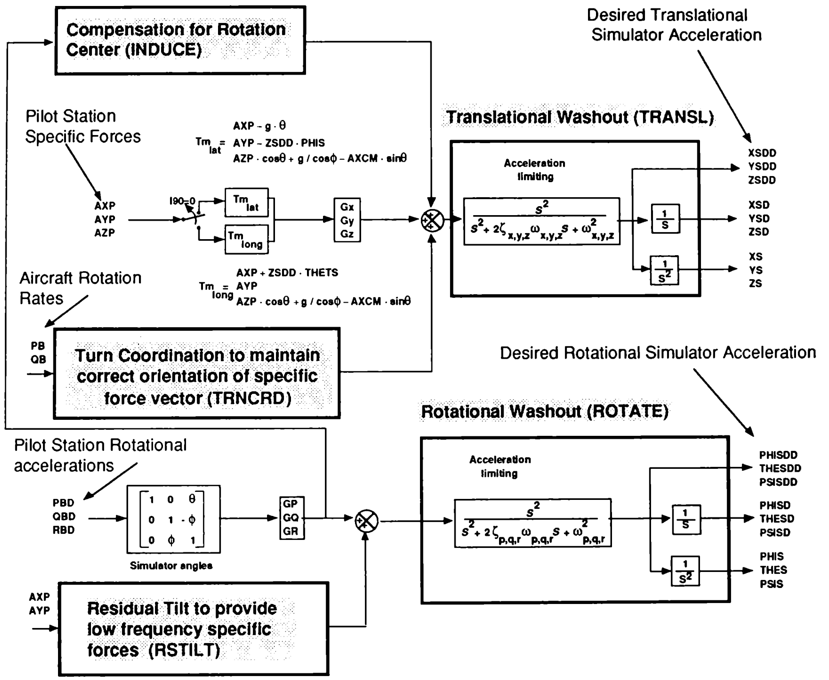

Compensation for Rotation Center (INDUCE)
Pilot Station Specific Forces
AXP
AYP
AZP
190=0
Tm_{lat} = AXP - g · θ
AYP - ZSDD · PHIS
AZP · cosθ + g / cosϕ - AXCM · sinθ
Tm_{lat}
Tm_{long}
Gx
Gy
Gz
Aircraft Rotation Rates
Tm_{long} = AXP + ZSDD · THETS
AYP
AZP · cosθ + g / cosϕ - AXCM · sinθ
PB
QB
Turn Coordination to maintain correct orientation of specific force vector (TRNCRD)
Desired Translational Simulator Acceleration
Translational Washout (TRANSL)
Acceleration limiting
$\frac{s^2}{s^2 + 2\zeta_{x,y,z}\omega_{x,y,z}s + \omega^2_{x,y,z}}$
$\frac{1}{s}$
$\frac{1}{s^2}$
XSDD
YSDD
ZSDD
XSD
YSD
ZSD
XS
YS
ZS
Desired Rotational Simulator Acceleration
Rotational Washout (ROTATE)
Pilot Station Rotational accelerations
PBD
QBD
RBD
$\begin{bmatrix} 1 & 0 & \theta \\ 0 & 1 & -\phi \\ 0 & \phi & 1 \end{bmatrix}$
Simulator angles
GP
GQ
GR
AXP
AYP
Residual Tilt to provide low frequency specific forces (RSTILT)
Acceleration limiting
$\frac{s^2}{s^2 + 2\zeta_{p,q,r}\omega_{p,q,r}s + \omega^2_{p,q,r}}$
$\frac{1}{s}$
$\frac{1}{s^2}$
PHISDD
THESDD
PSISDD
PHISD
THESD
PSISD
PHIS
THES
PSIS

Figure 1. Motion command computations

For this study, all software solutions to transfer functions used in the motion washout have been standardized from a variety of methods (typically Euler integration and knowledge of the governing differential equations) to a State Transition Matrix method. These two methods are referred to as the Euler Integration method and the State Transition method, respectively.

The output of an Euler integrator has a time of applicability (see next section) one half time step advanced from the input. During the implementation of a second-order high-pass filter, two integrations are necessary. Thus, with the Euler Integration method, and an acceleration input to the filter, the state vector is applicable at three different times: acceleration at the input time t, velocity at t + T/2, and position at t + T.

The State Transition method provides a state vector with all states applicable at time t. The State Transition method also provides highly accurate discrete representation of the intended analog transfer function.

Time of Applicability

In a continuous system, such as an analog filter, the output is concurrent with the input. Therefore, for analog systems, the concept of "time of applicability" is trivial. However, in a discrete process, such as a digital algorithm simulating an analog filter, the output may or may not be applicable at the same time as the input. The choice of the time of applicability is not arbitrary; it will affect the relative phase of the output.

Certain discrete algorithms advance the time of applicability; others do not. Those that do are useful in the implementation of systems involving closed control loops: if the algorithms used in a control loop are chosen such that a single integration time step of phase advance results (in addition to the phase of the continuous system modeled), then the feedback will have the same time of applicability as the input, because the feedback will have been computed during the previous iteration. If the transport delay in the feedback loop is not accounted for in this fashion, the summation will be improper, and will introduce both phase and magnitude error at the output of the closed loop.

State-space algorithms, which may be configured to produce an advance in the time of applicability, generally require some assumption to be made about the behavior of the input during the future interval (extrapolation). Common assumptions include the zero-order data hold (i.e., the input is assumed to remain constant during the interval) and the first-order data hold (i.e., the first derivative of the input is assumed to remain constant during the future interval). If the algorithm is derived from one of these two assumptions, it will exhibit a phase advance equivalent to a time advance of T/2 for the zero-order hold, or T for the first-order hold, where T is the cycle time. On the other hand, the configuration of a triangular data hold, which assumes that the first derivative of the input was constant during the past interval (interpolation), results in an output concurrent with the most recent input. Therefore, we have concluded that (1) the triangular data hold should invariably be used unless the time of applicability must be advanced; (2) around any closed loop, the time of applicability should be advanced exactly one time step; and (3) all variables entering a summing junction must be applicable at the same time.

The Euler Integration method can be derived from the zero-order data hold assumption applied to a single integration. It therefore has the T/2 time advance characteristic described above. When used to compute a second-order filter, however, two integrations are required, resulting in the velocity being advanced by T/2 with respect to an acceleration input, and the position being advanced by T. Although the position feedback therefore has the correct phase, since it is fed back during the next computation cycle, the velocity feedback has a net delay of T/2 at that same time. This delay causes both phase and magnitude errors, which are compounded by the fact that the output should have been concurrent with the input.

State Transition Method Software

The software used in the VMS facility for the State Transition method; the XFR system, allows a choice of the data hold. The theoretical basis of this material has been published [5]. Transfer functions of arbitrary polynomial order are accommodated, along with time-varying coefficients.

The triangular data hold option of the XFR software is especially useful, because it delivers complete solutions to equal-order transfer functions. Because this option delivers all states with an identical time of applicability (the most recent input time), it will also be quite useful in future research projects in the subject of delay compensation.

Performance of State Transition Versus Euler Integration Method

Filter implementation via Euler Integration is straightforward. For a second-order high-pass filter:

$$\frac{X_{out}}{X_{in}}(S) = \frac{S^2}{S^2 + 2\zeta\omega S + \omega^2} \qquad (1)$$

Which may be manipulated to obtain the output as a function of its integrals and the input:

$$X_{out} = X_{in} - \frac{2\zeta\omega X_{out}}{S} - \frac{\omega^2 X_{out}}{S^2} \qquad (2)$$

When the input and output of Equation (1) are accelerations (e.g., $\ddot{\phi}_{in}$ and $\ddot{\phi}_{out}$) Equation (2) may be written as a time domain equation:

$$\ddot{\phi}_{out} = \ddot{\phi}_{in} - 2\zeta\omega\dot{\phi}_{out} - \omega^2\phi_{out} \qquad (3)$$

In the case of Euler Integration, consider the mix of applicable times: We have $\ddot{\phi}_{in}$ applicable at the beginning of the current frame, ϕ_{out} applicable at the end of the current frame (two Euler integrations), and $\dot{\phi}_{out}$ applicable halfway between (one Euler integration). Hence, $\ddot{\phi}_{out}$ is calculated incorrectly, and the values of ϕ_{out} and $\dot{\phi}_{out}$ obtained by integrating $\ddot{\phi}_{out}$, are also incorrect.

The VMS is a six degrees-of-freedom simulator. The hardware controlling four of the six axes requires only a position command (however, the position command is normally provided with some lead compensation in the form of velocity and acceleration information). Of the other two axes, one requires (at the hardware level) both position and velocity commands, and the other requires position, velocity, and acceleration commands. Since all driven axes require a position command, the discussion (later in this section) of the differences between the State Transition method and the Euler Integration method focuses on the calculation of the position command resulting from the filtered pilot station acceleration.

To verify that the motion drive software performs correctly, a uniform random noise generator was used to provide a stimulus to the various filters. Both the stimulus and response were recorded, and frequency response estimations were made (via Fast Fourier Transforms) from the recorded data, using system identification software based on [6].

Figures 2 and 3 show the frequency response of Equation 1 followed by a double integration, in effect, a second-order low-pass system. The second-order low-pass filter is the result of an acceleration input to Equation 1, which results in an acceleration command, which is then integrated twice to obtain a position command. In both figures, the solid line is the experimentally determined frequency response of the filter. The broken line is the theoretical (Laplace transform) frequency response of the filter.

Figure 2 shows the frequency response of the roll axis washout filter when implemented with the Euler Integration method. Figure 3 shows the same filter using the State Transition implementation. Note the phase lead that is generated by the time advance resulting from the Euler Integration algorithm. Figures 2 and 3 were produced using a corner frequency (ω) of 0.7 rad/sec and a damping ratio (ζ) of 0.707, which values are representative of those used for simulations in the VMS.

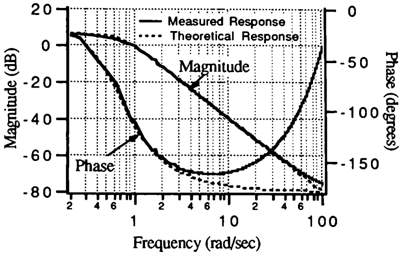


Figure 2. Euler Integration method frequency response of roll washout. Position response to an acceleration command

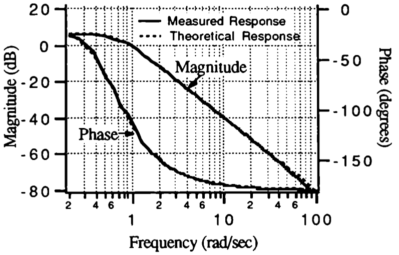


Figure 3. State Transition method frequency response of roll washout. Position response to an acceleration command

Effect of Algorithm on Lead Compensation

Pilot comments during a subjective evaluation of the State Transition and Euler Integration methods suggested one major difference in the cues provided: the State Transition method provides smoother motions than does the Euler Integration method. The frequency response of the washout system alone is not sufficient to explain this pilot observation. However, quantitative data to support the pilot claims are obtained when the frequency response of the combined effects of washout and lead compensation are considered.

The frequency response of the washout filter (Equation 1 with an acceleration input and output) is shown in Figures 4 and 5, for the Euler Integration and State Transition methods, respectively. Note that, in spite of the phase error present when the position is calculated with the Euler Integration method (Figure 2), the acceleration output of the filter (Figure 4) is not significantly affected. Figure 6 shows the combined washout and lead compensation for the roll axis. The pitch, yaw, and longitudinal axes use an identical scheme. The vertical and lateral axes operate differently; they are not discussed here, except to mention that feedforward is implemented in both hardware and software, rather than only in software as is done with the rotational and longitudinal axes. In Figure 6, the desired simulator state vector is shown between the washout and the lead compensation. The lead compensation (feedforward) adds small amounts of the desired velocity and acceleration to the desired position to provide a position command that will reduce the phase lag and attenuated amplitude inherent in the motion hardware. The acceleration which results from the compensated position command is calculated ("Second Derivative of Position Command" in Figure 6) to determine the frequency response of the combined washout and lead compensation. Figures 7 and 8 show the frequency response of the combined washout and lead compensation for the Euler Integration and State Transition methods, respectively. The frequency response plots shown in Figures 7 and 8 were generated using the "Acceleration Input to Washout Filter" and the "Second Derivative of Position Command" (Figure 6) as command and response.

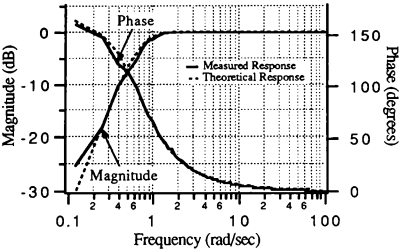


Figure 4. Euler Frequency response of roll washout. Acceleration response to an acceleration command

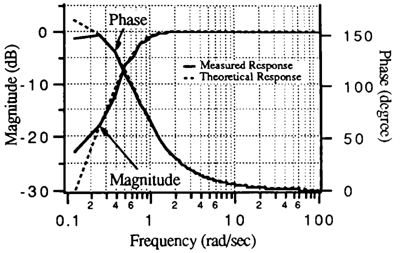


Figure 5. State Transition method frequency response of roll washout. Acceleration response to an acceleration command

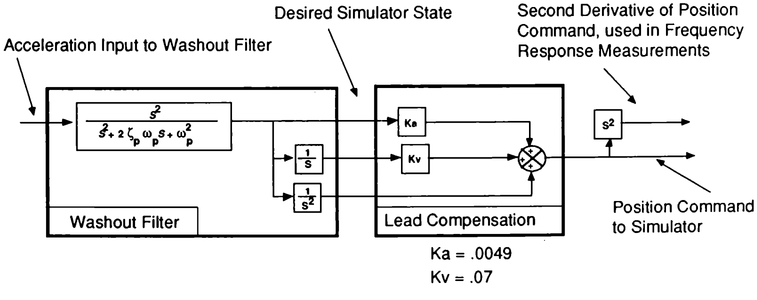

Desired Simulator State
Acceleration Input to Washout Filter
Second Derivative of Position Command, used in Frequency Response Measurements
$\frac{s^2}{s^2+2\zeta_p\omega_p s+\omega_p^2}$
$\frac{1}{s}$
$\frac{1}{s^2}$
Ka
Kv
s^2
+
+
+
Washout Filter
Lead Compensation
Position Command to Simulator
Ka = .0049
Kv = .07

Figure 6. Roll axis washout and lead compensation

The frequency response plots in Figures 7 and 8 show the expected behavior of a second-order high-pass filter from low frequencies up to about four radians per second; above four radians per second, the feedforward begins to dominate; by about 20 radians per second the response is that of two differentiations. From about 12 radians per second to about 60 radians per second, the Euler Integration method shows a gain of about 3 dB (40%) over the theoretical response. The State Transition method shows no increased gain.

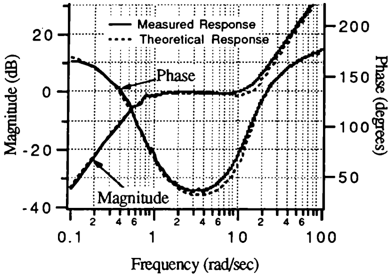


Figure 7. Washout and feedforward acceleration response using the Euler Integration method

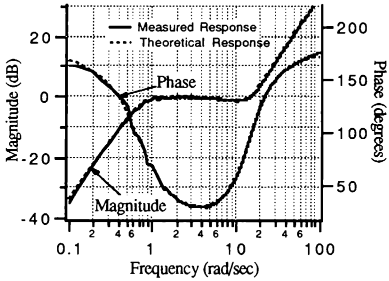


Figure 8. Washout and feedforward acceleration response using the State Transition method

The high frequency gain error, which occurs only after the lead compensation, is the only significant, quantitative, difference that has been observed when comparing accelerations, between the Euler Integration and the State Transition methods. Experienced VMS test pilots described the motion provided by the Euler Integration method as "abrupt" and "jerky," when compared back to back with the State Transition method. The gain amplification observed at high frequencies can explain these comments.

Piloted Evaluation

In addition to the quantitative data already discussed, subjective data was collected from pilots who flew the VMS during a brief evaluation to compare the new motion drive software using the State Transition method with the old motion drive software using the Euler Integration method. Measured acceleration differences at the pilot's location were not available during the evaluation.

The simulation used in this evaluation was a generic (stability derivative) fixed wing aircraft model. The four pilots who participated were either NASA test pilots employed at Ames Research Center (ARC), or Army test pilots assigned to ARC.

The evaluation called for each pilot to perform the five tasks listed below. To obtain as close a comparison as possible, each task was performed with both sets of drive software before proceeding to the next task.

The following tasks were selected for the evaluation:

1. Roll steps. From wings level, roll right to 30 degrees; hold approximately one second, return to wings level. Repeat for left roll.

2. Pitch steps. From level flight, pitch up 15 degrees; hold one second, return to level flight. Repeat for pitch down.

3. Roll reversals. Smooth, continuous left and right rolls to roll angles of approximately +/- 30 degrees. A period of four to six seconds was maintained.

4. Pitch reversals. The same as roll, but with pitch angles of +/- 15 degrees.

5. Throttle steps. From level flight, throttle was increased approximately 20%, held for five to ten seconds, and returned to the original setting.

Figure 1 illustrates the design of the motion washout, as well as a number of gains, damping ratios, and corner frequencies which may be adjusted to optimize the motion system performance. During the piloted evaluation, the same gains, corner frequencies, and damping ratios were used in the two different sets of drive software. During the evaluation, the pilots were informed when one set of motion drive software was replaced with the other. However, they did not know which system they were using.

The pilots were asked to evaluate whether or not the two methods provided different cues and, if the cues were different, how they were different. Since the aircraft model in use did not represent an actual aircraft, the pilots were not asked which method produced a "better" simulation; however, they were asked whether they preferred one method to the other, and if so, why.

The evaluation was broken down further into the following categories: onset cues, steady-state cues, and phase of motion with respect to the visual scene. General comments were also solicited.

Three of the four pilots preferred the motions provided by the State Transition method to those provided by the Euler Integration method. The one pilot who preferred the motions provided by the Euler Integration method did not complete the tasks; he also had the least experience with the VMS. The most common comment was that the motions produced by commands calculated using the State Transition were smoother than motions produced from commands calculated with the Euler integration method.

Conclusions

The State Transition method has been shown to more accurately implement the analog transfer functions specified in the washout design. In addition, pilot comments during this preliminary study indicate that, in the VMS facility, vestibular cues obtained from a consistent state vector that has not been advanced one cycle (25 milliseconds, in this case) are preferable to those obtained from an inconsistent state vector with a one cycle advance in position and a half cycle advance in velocity.

Pilot comments describing the motions produced by commands calculated with the Euler Integration method as "jerky" may be explained by the excessive high frequency gain shown in Figure 7. The smoother motions produced via the State Transition method prompted one experienced VMS test pilot to comment that "the motion has never felt better."

References

1. Sinacori, J.B., "A Practical Approach to Motion Simulation," AIAA Paper 73-931, September, 1973.

2. Bray, R.S., "Initial Operating Experience with an Aircraft Simulator having Extensive Lateral Motion," NASA TM X-62,155, May, 1972.

3. Conrad, B. and Schmidt S.F., "A Study of Techniques for Calculating Motion Drive Signals for Flight Simulators," NASA CR-114345, July, 1971.

4. Schmidt, S.F. and Conrad, B., "Motion Drive Signals for Piloted Flight Simulators," NASA CR-1601, May, 1970.

5. McFarland R.E. and Rochkind A.B., "On Optimizing Computations for Transition Matrices," IEEE Transactions on Automatic Control, Vol. AC-23, No. 3, June 1978.

6. Bendat, Julius S. and Piersol, Allan G., "Engineering Applications of Correlation and Spectral Analysis." John Wiley and Sons, 1980.

AIAA-92-4159-CP

AUTOMATING SIMULATOR OPERATIONS

Rebecca J. Jacobs
Operations Support Contract/Barrios Technology, Inc.

C. Timothy Featherston
The MITRE Corporation

Johnson Space Center
Houston Texas, 77058

Abstract

As the complexity of flight simulators has increased, the time required to support these simulators has increased as well. NASA's experience with flight simulators has shown that the manual procedures used in the past have been manpower intensive. Vital and expensive training time is often lost or degraded while problems are being worked. The Space Station Training Facility will be used to train astronauts and ground support personnel to perform the tasks which will be required to assemble, operate, and maintain the Space Station Freedom. It will contain computational systems, a number of crew stations, onboard computers and flight software, and instructor stations. These components will be apportioned into one or two full-task man-in-the-loop simulations. To configure the components for training, and to monitor and control the components during realtime will require an Operations Support System containing an integrated set of tools for system configuration, and realtime data monitoring, analysis, and control.

Nomenclature

DMS	Data Management System
GFE	Government Furnished Equipment
JSC	Johnson Space Center
NASA	National Aeronautics and Space Administration
OSS	Operations Support System
OST	Operations Support Team
SaC	Status and Control
SCT	System Configuration Tool
SSCC	Space Station Control Center
SSTF	Space Station Training Facility

Introduction

As the complexity of flight simulators has increased, the time required to support these simulators has increased as well. The National Aeronautics and Space Administration's (NASA's) experience with training facilities such as the Shuttle Mission Training Facility has shown that the manual operations procedures used in the past have been manpower intensive. There are many reasons for this. Since the user is not a trained maintenance person, they may not be able to adequately describe the problem. In some cases, the user may not remember when and under what conditions the problem occurred. The data required to reproduce the problem may not have been saved. This often results in lost or degraded training time while problems are being resolved. This makes it necessary to rely upon a select group of experts to diagnose and correct problems.

Industrial experience has shown that human involvement in problem diagnosis and operations can be reduced. Systems exist to assist operators in quickly isolating and diagnosing faults(2), and in operating networks(7) or spacecrafts(6). To improve simulator operations, the Space Station Training Facility (SSTF) currently being constructed at NASA's Johnson Space Center (JSC), will contain an integrated set of automated operations tools for system configuration, and realtime data monitoring, analysis, and control.

Background

The SSTF will be the primary Space Station Freedom Program facility for training crew members and ground support personnel in the normal and contingency operations of the Space Station Freedom. The SSTF will contain computational systems, a number of independent crew station modules, onboard computers and flight software, and instructor stations. Figure 1 shows the major components of the SSTF. Table 1 lists the purpose of these major components. All SSTF components will be apportioned into one or two man-in-the-loop simulation sessions. Networks will connect the components together.

To configure the components for training and to monitor and control the the components during realtime, the SSTF will require an Operations Support System (OSS). The OSS will execute the operations software, store operations data, and provide access to operations documentation. The OSS will be the primary repository for SSTF simulation loads. During full-task simulations, the OSS will be used to configure the SSTF components into the simulation sessions based on user specified requests. For example, a user may request the habitation module, laboratory module and cupola with increment flight software to be used in the upcoming flight. These sessions will be initialized to one of the standard configurations: standalone, combined, integrated and joint integrated.

The standalone configuration will allow training to be conducted within any single module independent of the other modules. The combined configuration will allow training to be conducted with a full-task session consisting of either one, or a combination of modules. The integrated configuration will allow full-task training to be conducted with a session consisting of either one, or a combination of SSTF modules configured to interface with the Space Station Control Center (SSCC). A joint integrated configuration will include the SSCC, other JSC facilities, and, through the SSCC, other NASA centers. A typical session will utilize four hours. A training session can and will be used to train a single student in the operation of a single system. On the other hand, it will also be used to train an entire crew to perform a coordinated operation (e.g., a reboost or docking).

The SSTF, including the operations tools, will be developed and maintained in a series of incremental releases. This paper describes the complete capabilities of the OSS as expected at the time of the final delivery.

Operations Concept

The requirements for the SSTF OSS were derived from utilization studies(4) and the facility operations plan(3). The utilization studies defined the SSTF users, their purposes for using the SSTF, and how often and for how long they will use it. The operations plan describes the activities required to prepare the SSTF for simulations and the composition of the Operations Support Team (OST) who will perform these activities. This information forms the basis of the SSTF operations concept.

The SSTF is expected to be operational throughout the life of the Space Station Freedom Program. During this 30 year period, the facility will be operational 24 hours a day, 7 days a week. To achieve the required level of availability, operations support must be proactive. That is, operations support must anticipate problems before the problems occur. To accomplish this, automated capabilities will be designed into the SSTF systems to monitor anomalies and take corrective active to prevent these anomalies from becoming serious problems affecting the facility users. These automated capabilities, in turn, will be monitored and controlled by the OST.

The OST will be responsible for configuring the facility to meet the needs of users. Each user session will be supported by the OST, who will be responsible for the configuring the simulation session. This includes the execution of tests to verify the integrity of the session and the integration of the session with any external facilities (e.g., the SSCC). The OST will also assist the user in the operation of a session. This assistance can include any or all of the following activities: 1) custom modifications to the configuration of the session, 2) problem identification and documentation 3) coordinating problem resolution, 4) restoration of the session after repair.

From the OST point of view, a simulation session begins with the identification of the configuration of items the session requires. The OST will, in most cases before the user is present, retrieve the information which defines the desired configuration. Based upon this information, the OST will determine the status of the configuration items (i.e., hardware, software, and data). As the items become available, the OST will begin to integrate them into the desired configuration. This will include the

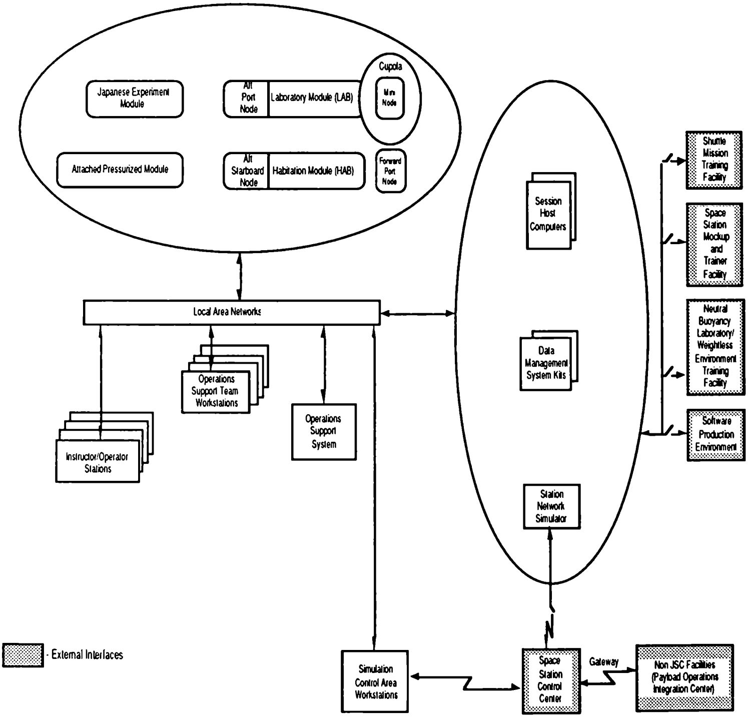

Cupola
Japanese Experiment Module
Aft Port Node
Laboratory Module (LAB)
Min Node
Attached Pressurized Module
Aft Starboard Node
Habitation Module (HAB)
Forward Port Node
Local Area Networks
Operations Support Team Workstations
Operations Support System
Instructor/Operator Stations
Session Host Computers
Data Management System Kits
Station Network Simulator
Shuttle Mission Training Facility
Space Station Mockup and Trainer Facility
Neutral Buoyancy Laboratory/ Weightless Environment Training Facility
Software Production Environment
▨ - External Interfaces
Simulation Control Area Workstations
Space Station Control Center
Gateway
Non JSC Facilities (Payload Operations Integration Center)

Figure 1. Space Station Training Facility Components and Interfaces

Table 1. SSTF Components

Component	Purpose
Operations Support System	Allows the OST to support training activities, to configure and operate the SSTF
Operations Support Team Workstations	Allow the OST to use the operations support capabilities to operate and manage the SSTF
Cupola/Mini-Node	Simulates the functionality of the cupola for proximity operations and robotic manipulation training and simulates the functionality of a node system
Forward Port Node	Simulates the functionality of a node system
Laboratory Module and Aft Port Node	Simulate the functionality of the laboratory module and a node systems module
Habitation Module and Aft Starboard Node	Simulate the functionality of the habitation module and a node systems module
Japanese Experiment Module	Simulates the functionality of the pressurized module and the interface to the space station core systems
Attached Pressurized Module	Simulates the functionality of the Attached Pressurized Module and the interface to the space station core system
Station Network Simulator	Provides the capability to integrate the SSTF with the Space Station Control Center (SSCC); simulate the network between the station and the control center
Session Host Computers	Provide Automatic Data Processing Equipment for the sessions
Data Management System (DMS) Kits	Provide a dedicated onboard computer for each session host computer for execution of flight software
Instructor/Operator Stations	Provide the hardware and software tools necessary to interface and interact with the simulation models to provide effective flight and ground crew training for the SSTF
Simulation Control Area Workstations	Allow the instructors to monitor the SSCC
Local Area Networks	Provide a realtime Local Area Network for time-critical simulation-unique traffic and a general purpose Local Area Network for non-time critical message traffic

execution of hardware diagnostics on equipment items which are not currently in use. Any problems or anomalies encountered during this process will be documented and provided to the user when they arrive.

If there are no unresolvable problems, configuring the simulation session will continue. At this point, the user may direct the OST to tailor the predefined configuration of the simulation session to include any unique user software and data. Once the configuration is complete, the OST will initialize the simulation session and turn it over to the designated user.

During a simulation session, the OST will monitor the operation of the session and perform any special user requests (e.g., integration with external facilities). The OSS will monitor the status of the session and inform the OST of anomalies or problems. If a problem is detected during the session, the OST will use the data collected by the OSS to analyze the problem. The OST will use the OSS to determine a solution or work around for the problem and restore operability to the user.

When a simulation session is terminated, the status of the session configuration items is automatically updated so that other future simulation sessions can begin the session start process described above.

System Configuration Tool (SCT) and Status and Control (SaC) Descriptions

The above operations concept requires centralized control, early detection and quick recovery of problems to maximize the time available for training. The requirements defining the capabilities of the operations tools[5] were developed to meet this operations concept. Since development of those requirements, the initial system designs and a proof-of-concept prototype have been developed. This paper describes the two most complex pieces of the OSS: the System Configuration Tool (SCT) and Status and Control (SaC).

Before training can begin, the selected hardware and software components must be connected and verified. The SCT will assist the OST in performing these functions. Before the selected hardware components are connected or the software loaded, the SCT will verify the current status (e.g., available, in use, under repair), compatibility, and consistency of the selected components of the predefined session. Once the selected configuration is active, the SCT will be able to change the configuration by adding or removing selected components.

Prior to initiation of realtime, the SCT will allow the user to select which hardware components should be included in an upcoming session and which software load (e.g. flight increment) is desired. The SCT will then determine which software components (e.g. flight software) must be loaded into the selected hardware. The user will have the capability to select from a menu of existing configurations and modify them to suit the current needs, or build a new configuration. Prior to session initialization, a graphical user interface will allow the OST to interactively allocate facility resources (e.g., a module or a network) to a simulation session. The OST will do this by moving icon representations of SSTF components into or out of the session groups.

Once the configuration has been selected, the SCT will load the hardware components with the correct software loads. The SCT will then query SaC to determine the availability and health status of the required components. Via the graphical user interface, the OST will be informed of this status and will be asked to confirm continuation of the requested configuration. Once confirmed, the information will then be stored for use by other OSS components (e.g., the productivity monitoring tool).

During realtime, the SCT can add or delete session components without disrupting training in process. This occurs either when requested by a user or when SaC detects a problem and automatically requests a component change. If the addition or deletion of a component is requested, the SCT will be invoked to repeat the procedure described above.

During realtime, SaC will collect data from all software and hardware components used in the session, including the DMS kits, flight software and other operations tools (e.g. logging or event tracing). SaC will correlate the collected data with the simulation activities (e.g. active malfunctions, simulation mode, etc.) and provide a comprehensive status to the OST. If needed, SaC will automatically initiate diagnostics to collect data on hardware problems. SaC will report all problems to the OST and offer options for correcting those problems. Using expert systems, SaC will be capable of automatically correcting problems or adjusting simulation performance parameters. As a safeguard, the automated control capability can be over ridden by the OST. SaC will also provide the capability to analyze the collected data to perform trend analysis and what-if scenarios.

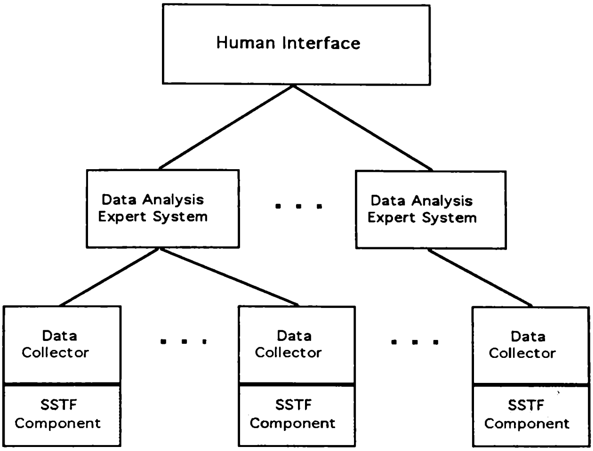


Figure 2. Status and Control Hierarchy

Figure 2 shows the SaC hierarchy. Raw data will be collected directly from the SSTF components by SaC data collectors running co-resident on the components. The SaC data collectors will be designed so that the performance profiles are as constant as possible, to minimize fluctuations in the simulation. This requires cyclic collection of a standard set of data. Collecting a consistent set of data has the added advantage of providing a data history in the event a problem occurs. For this reason, defining in advance the type of data needed is crucial. All subsystems being developed for the SSTF will supply SaC with some type of data. SaC will collect outputs from diagnostic routines,hardware status data, software status data, simulation model error messages, error messages from the realtime support system software, resultant data from SaC diagnostic activities, and data from other OSS components.

The data collectors will provide the data to a series of data analysis expert systems. These expert systems will run on a group of dedicated workstations performing anomaly and problem detection and categorization based on system rules. The rules and algorithms may consist of regression analysis, single limit checks or multiple limit checks performed by more than one expert system. To perform multiple limit checks, the expert systems will share data. Some shared data will be used by system rules which analyze interfaces between the components.

The human interface will format the analyzed data for viewing by the OST. The human interface will perform all display processing. To avoid overwhelming the OST with data, the human interface will perform message filtering. Displays will present data as a combination of graphic displays and text messages. Text messages noting problems will include a criticality level and description of the problem. The human interface will perform limit checks on user defined variables. It will allow the OST to check or adjust these limits while the simulation is running. The human interface will also accept and route user commands for execution at the appropriate component.

Other Automated Operations Tools

Supplementing and related to SaC and the SCT will be an integrated set of operations support tools. The OST will use some of these tools to prepare for a simulation; some will be used during the simulation and others will be used after the simulation. Online tools will provide configuration management, data logging and event recording, programmable event capability, centralized system debugging, and online documentation. Offline tools will provide data delogging, system analysis, documentation management, training history, and session productivity reporting.

Within the SSTF environment, the configuration management tool will track and control products such as software executables and data, and simulation generated products. The configuration management tool will provide the hardware and software compatibility analysis for the SCT. It will also assure system integrity by controlling user access to baselined products.

Simulation event recording will provide a time ordered record of simulation events from multiple sources. From these recorded events a user will be able to extract events for replay, and create or modify an executable script. The scripting capability will include creation and editing of scripts offline. Related to event recording is the logging capability which will allow users to specify data values to be logged, the duration and rate of logging. The data will be analyzed offline using the data delogging capability. During problem analysis, the OST will activate debugging and diagnostic tools from OST workstations. Past experience has shown the need to collect and store data concerning the productivity and utilization of simulation sessions. This will be provided by a productivity monitoring tool.

System analysis tools will provide the capability to interactively model system loading and compare the results to the actual system loading. A documentation management tool will provide the capability to view, print, manage, and report on documents from multiple libraries A training history database will track maintenance and operations personnel training and certification levels.

Issues

Technology and time have solved the traditional problems associated with automation. Technology has provided a wide array of powerful computer platforms and software. Time has provided us with the knowledge of which functions should be automated and the experience required to implement and test those functions. However, these solutions have also brought with them new problems.

The SSTF architecture, as described in the background section of this paper, will be both logically and physically distributed. The hardware elements of the architecture will be spread across several floors of the building which will house the facility. The software architecture will be logically distributed over a number of computers. As a result, the automated operations support system must also be both physically and logically distributed.

If this system architecture is to satisfy the requirement for proactive problem resolution and to meet the demanding timing requirements of a realtime flight simulation, then individual elements of the system must be capable of autonomous decision making. However, in order to make autonomous decisions correctly, the element may require extensive knowledge of the state of the other elements in the system. A tradeoff must be made between the type of decisions or actions which can be made locally, and the burden that transmitting the information required to make that decision places upon the communications resources.

The SSTF will contain commercial hardware, government furnished equipment (GFE), and custom built hardware. The OSS is required to collect information about and maintain control of all of this hardware. However, this may not be feasible in all cases. While it may be possible to provide data and control access into the design of the custom hardware, it is not likely that either commercial or GFE hardware will provide this kind of access.

Some commercial hardware, in particular communications hardware, provides access to internal status and some external control of built in diagnostic functions. However, most commercial hardware vendors do not provide this type of electronic access. The requirements of the OSS have been and will continue to be part of the requirements of each commercial procurement. However, these requirements cannot be the sole criteria upon which a procurement is based. As a result, the implementation of the OSS will be compromised by the amount of information and control provided by commercial hardware selected for the SSTF.

The SSTF architecture will contain a number of hardware items which are provided as GFE. In particular, the SSTF will contain a number of functionally equivalent flight computers (DMS kits) and signal conditioning devices. It is unlikely that this equipment will provide the access required to support the OSS.

Like the hardware, the SSTF software will be a mix of commercial, GFE, and custom software. It is unlikely that commercial software vendors will provide the information required to monitor or modify the operation of their software in realtime. The GFE software suffers from a similar problem. In all likelihood, this software will be delivered to the SSTF as a binary image suitable for execution in the functionally equivalent flight hardware. It is only in the case of the custom built software that the developers will have access to the information and controls required to implement the full potential of the OSS.

If the OSS is to assist the OST in the identification of problems, it must not only collect and display information about all of the items in the simulation, it must also have some knowledge about the expected behavior of these elements. If it is to assist the OST in resolving problems, then it must be cognizant of what actions are possible and the effect that each action will have. These are the properties of an expert system(1). The collection, verification, and validation of the knowledge base for this expert system will be very challenging. These difficulties will be compounded by the incremental development of the SSTF and the need of the OST to include temporary rules to support temporary operational solutions to problems.

A number of efforts are underway to address the issues described in this section. The issues relating to the development and operation of the expert system are being addressed during the design and with the development of a prototype. This prototype has been developed to assess the issues relating to the interface with the OST. A number of designs are being considered to address the issues of the distributed system architecture and autonomous decision making.

As discussed above, the constraints placed upon the implementation by the use of commercial and GFE hardware and software items are being addressed. In the case of the commercial items, the requirements of the OSS are being included in the technical requirements and selection criteria for the procurements of these items. Issues with GFE hardware and software items are being discussed in a series of ongoing technical exchanges with the providers of these products.

Conclusions

The requirements and operations concept for automating the operations of a facility are not new. Every type of production facility has the need to increase efficiency and reduce the amount of time spent resolving problems. This is particularly true in facilities like the SSTF which are constantly impacted by high rates of change and wide fluctuations in both the type and amount of utilization.

To meet these needs, the OST have defined a set of requirements for an operations support system. The system will assist the OST in almost every facet of the facility operations, management, and maintenance. The task of automating operations in a facility like the SSTF is an ambitious one. The issues and risks which are currently known are being addressed. The most serious issues deal with access to the necessary data and controls for commercial and GFE hardware and software products.

The system has completed the preliminary design phase of development. The first increment of the system is scheduled to become operational in March, 1995. The full operational capability of the automated operations support system will be online in the SSTF in June, 1999.

References

1. A. Barr and E. Feigenbaum, "The Handbook of Artificial Intelligence, Volume 1", Heuristic Press, Stanford California (1981)

2. K. Howlin, J. Weissert, and K. Krantz, "MOORE, A Prototype Expert System for Diagnosing Spacecraft Problems", Proceedings of the 1988 Goddard Conference on Space Applications of Artificial Intelligence, NASA Conference, Publication 3009, p. 175-190 (1988)

3. NASA , "Space Station Training Facility (SSTF) Facility Operations Baseline Operations Plan, JSC-36010", Johnson Space Center, Houston, Texas (1991)

4. NASA, "Space Station Training Facility (SSTF) Resource Loading Projections, JSC-36040", Johnson Space Center, Houston, Texas (1992)

5. NASA, "Space Station Training Facility (SSTF) User Detailed Functional Requirements Document, JSC 24454", Johnson Space Center, Houston, Texas (1992)

6. J. Rash, Y. Wong, and J. Cieplak, "A Rule Based Systems Approach to Spacecraft Communications Configuration Optimizations", Proceedings of the 1988 Goddard Conference on Space Applications of Artificial Intelligence, NASA Conference, Publication 3009, p. 141-153 (1988)

7. W. Wilkinson, N. Happell, S. Miksell, R. Quillin, and C. Carlisle, "Achieving Real-Time Performance in FIESTA", Proceedings of the 1988 Goddard Conference on Space Applications of Artificial Intelligence, NASA Conference, Publication 3009, p. 191-205 (1988)

AIAA-92-4160-CP

SIMULATION SOFTWARE FLOWDOWN IN THE AIRCRAFT DESIGN PROCESS: PROBLEMS AND SOLUTIONS

L. M. Landry, Jr.*
Dorothy M. Baldwin**
General Dynamics Corporation, Fort Worth Division
Fort Worth, Texas

Abstract

Software models of aircraft systems are typically developed several times in the design of Military aircraft. As many as four different sets of software which model the aircraft may be developed, resulting in additional cost and schedule, and requiring additional validation at each step. This paper addresses this process, discusses some of the impediments to software reuse in the design process, gives examples of software reuse in recent programs, overviews recent developments which offer potential for improvements, and gives examples of improved processes.

INTRODUCTION

Flight simulators have become an essential part of military aircraft development. Closely following increases in digital computer technology, the complexity and capability of flight simulators has steadily increased to the level found in modern Full Mission Simulators. Military aircraft companies have made large capital investments in Engineering Flight Simulators (EFS) to support the design of their aircraft, and to a large part agree that these simulators have played a critical role in successful aircraft design.

Reference 1, "Air Combat Simulation and its Role in the Aircraft Development Process", discusses the phases of military aircraft design and development which utilize simulator technology at General Dynamics, Fort Worth Division. These phases are shown in Figure 1, which describes the process in which aircraft systems are evaluated in the EFS in both software "candidate configuration" and "flight hardware and software" forms. This has been found to dramatically reduce both bench test and flight test time required to validate Flight Control designs.

The military has also grown very reliant on flight simulators for aircrew training. These Pilot Training Simulators (PTS) have become essential in training modern aircrews in increasingly sophisticated weapons systems.

At the heart of the flight simulator are the mathematical models and software codes which represent the aircraft being designed. These models include representations of aerodynamics, flight controls, atmosphere, landing gear and actuation systems.

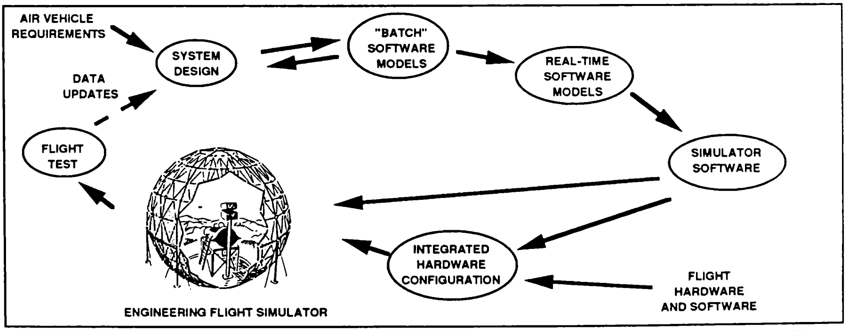

AIR VEHICLE REQUIREMENTS
SYSTEM DESIGN
DATA UPDATES
"BATCH" SOFTWARE MODELS
REAL-TIME SOFTWARE MODELS
SIMULATOR SOFTWARE
FLIGHT TEST
INTEGRATED HARDWARE CONFIGURATION
FLIGHT HARDWARE AND SOFTWARE
ENGINEERING FLIGHT SIMULATOR

FIGURE 1. THE DEVELOPMENT CYCLE FOR FLIGHT CONTROL SYSTEMS USING FLIGHT SIMULATORS

* Engineering Specialist, Sr.
Senior Member, AIAA

** Engineering Project Manager
Member, AIAA

Other models represent the avionics, weapons, radar and navigation systems. Still other models, such as threats, targets, etc., are required in the simulator to represent the mission in which the aircraft is designed to fly.

This paper addresses the first type of models, which represent the aerodynamic and kinematic aspects of the aircraft. In the typical design cycle, models of these systems are developed in the pre-design stage, in the design stage in "batch" form for use in non real-time analysis of (for example) flight control systems, in real-time form in the EFS, and again in the PTS.

Historically, it has been common to find unique models and software in each stage of development (Figure 2). While partially a function of the different computational resources normally found in the design functions responsible for each phase, the need for unique models and software seems to have been more dependent on the different design cultures in each function. The proliferation of models and software in the design stages contributes to problems in the overall organization of models and data in the design of the aircraft. These problems manifest themselves in the difficulties of configuration management, the necessity of redevelopment and revalidation of models at each design stage, and the resultant effects in cost and schedule.

A particular problem surfaces in the last stage of the process, the design of the PTS. As well documented in the "Simulator Data Integrity Study" (Reference 2), the necessity to redevelop aircraft models for the PTS has been a strong contributer to delays in development. In addition to delaying the availability of the PTS for crew training (as much as 18 to 24 months after the fielding of the aircraft, per Reference (3)), there are also associated cost increases and life cycle costs related to maintaining the PTS concurrent with the aircraft.

CAUSES OF THE PROBLEM

In the areas of detailed design found in the aerodynamics and flight controls design areas, there is a logical connection with the EFS. Traditionally, large "batch" models are developed in this phase, and used to support detailed design. These models are also used to provide test data which is used to validate the flight simulator. Historically, software developed by flight controls and aerodynamic engineers for their design has, however, not been usable in the EFS, as software developed in the design areas could not readily execute on flight simulation computers in real time. This is due to the complexity of the models reflected in the software, such as large amounts of tables representing wind tunnel data, and functions in the software not consistent with real time execution, such as frequent interface with the operator during execution. As a result, flight simulator software has traditionally been developed separately from software utilized in design areas.

The problem is exacerbated in the PTS. Factors that inhibit reuse of EFS software in the PTS have included:

- Differences in software requirements between the PTS and EFS, such as ADA and Software development standard MIL-2167A.

- Different relationships between the EFS and PTS and their sources of requirements. The PTS develops a single product from a set of requirements representing an aircraft which has

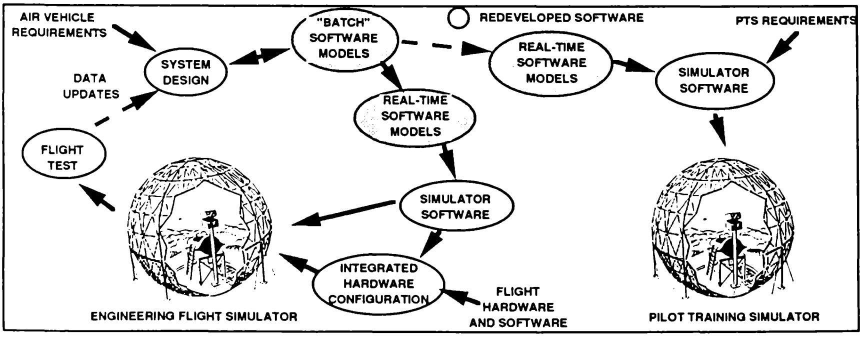


FIGURE 2. DEVELOPMENT OF EFS AND PTS SYSTEMS SHOWING REDEVELOPED SOFTWARE

been designed. The EFS is a tool utilized to develop this design, hence must continuously iterate on varying data representing an evolving aircraft design.

- Different formality of development processes and documentation result from the previous factor.

- Different functional requirements in the models developed in the PTS and EFS. For example, the PTS must replicate failure modes in subsystems which may not be present in the EFS.

- Different developers. The PTS has traditionally been developed by a company different than that which developed the aircraft. This factor centers on the difficulties related to transferring design data from the developer to the PTS contractor, and have been extensively addressed in Reference (2).

- Different levels of fidelity. For example, PTS developers have traditionally remodelled complex functions such as the aerodynamics model, in order to be able to execute on more affordable computer systems.

Reference (2) contains an excellent analysis of these and other factors.

EXAMPLES OF FLOWDOWN

The process of reusing of software developed in one stage of the design process in another stage is known as "flowdown". Incorporation of flowdown in the design process has long been recognized as a factor which could reduce the redundancy found in traditional processes, permitting cost reductions and schedule improvements.

While problems with software flowdown in the design process are significant, there are nonetheless examples of aircraft developers who have introduced improvements to the design process to reuse software.

The V-22 Osprey program at Bell Helicopter developed an EFS to support the design of the V-22. The core of this simulator utilized FORTRAN software which represented the aerodynamics and dynamics of the V-22.

This software was common between the Bell and Boeing simulators, and was used as the design baseline for the Operational Flight Trainer PTS. ADA software was redeveloped for the PTS, which made use of the mathematical models inherent in the GTR software.

This software was also reused in the U. S. Navy Simulators at Patuxent River, Maryland, to support Developmental and Operational Testing (DT/OT). Availability of this software across the spectrum of engineering and testing was a significant factor in the success of both programs.

McDonnell Douglas Training Systems has reused FORTRAN software modules from the McDonnell Aircraft EFS on some PTS systems. The military customer had flown the trained pilots on the EFS. His confidence in this software in the EFS resulted in acceptance of EFS software for the PTS. These two factors (same organization for the EFS and PTS, and close communication with the end user) were clear contributers to the success of the flowdown.

NEW DEVELOPMENTS WICH ENABLE FLOWDOWN

Three of the largest impediments to flowdown have been divergent processor requirements, divergent software processes and differences in the administrative and "cultural" aspects of the three nodes in the design process (design areas, EFS and PTS). Flight simulators have not been able to execute software developed in design areas due to real time constraints, PTS software development processes have required more formality than that normally provided in the EFS, and traditional organizations have been structured insufficiently to insure the incorporation of "downstream" requirements in "upstream" developments.

PROCESSORS

Traditionally, models developed in design areas were required to be redeveloped in the flight simulator due to real-time processing requirements. Further, the large and complex processors required in the EFS to provide flexibility and fidelity were too expensive to be appropriate to PTS products.

Historically, the EFS processors which compute the large aerodynamic models have consisted of special purpose array or pipeline processors, such as the FPS-5000 or the AD-100. In addition to the expense of these systems, they also have the drawback of not efficiently supporting the higher level languages (HOL) commonly found in design areas, such as FORTRAN, C or ADA.

Recently, microprocessors have been developed which are low cost and which support the HOLs. They have the critical capability of being able to execute classes of software developed in design areas (such as large, table-based aerodynamic models) in real time. The combination of HOL support, speed and low cost make them ideal to support flowdown.

SOFTWARE DEVELOPMENT PROCESSES

As discussed above, the differences in design goals between the EFS and PTS have led to divergence in documentation levels and formality of software development. This has undermined the applicability of EFS software to the PTS, and has been a factor in the requirement of new software for the PTS.

Recent trends in the aerospace industry have led to a renewal of emphasis on quality in Engineering processes. At General Dynamics, for example, one of the key areas of application of quality has been the software development process. While the need for different levels of formality has been recognized, General Dynamics standards have been developed which specify minimum levels of formality to be imposed on each level of software developed (deliverable, support of deliverable and non- deliverable). These processes are based on the MIL-STD 2167A process, and hence provide a basis for flowdown of software between the EFS and PTS.

ORGANIZATION MODELS

Although solutions to the above problems are becoming technically feasible, traditional "design cultures" are another factor which inhibits flowdown. The common planning and common design decisions which are critical to flowdown must happen very early in the program to ensure incorporation into the detailed designs of the three functions. Traditional administrative structures have inhibited this synergy.

One of the manifestations of "quality cultures" which are evolving out of new quality processes is the "natural work group" which recognizes that all functions which contribute to a product should interface on whatever level appropriate to that contribution. At General Dynamics, this approach takes the form of the "Integrated Product Team" (IPT), in which the contribution of participating functional areas is ensured sufficiently early in the design process enable an integrated design. This administrative synthesis offers significant potential for the design synergy necessary for the connection between the three areas which develop and utilize simulation software.

The IPT concept offers potential for flowdown which transcends the traditional "reuse" concept. The existence of a multi-disciplinary team which designs and develops software to meet potentially divergent requirements reinforces the goals of "concurrent" engineering. In this improved engeering process, the concept of "flowdown" leads to the concept of "crossflow" in which both EFS and PTS areas benefit from software designed and developed in common.

IMPROVED PROCESS

As explained above, recent advances in processor technology, quality initiatives in software processes and Integrated Product Teams open a new avenue of improved flowdown. At General Dynamics, new programs are capitalizing on advances in these areas.

F-16 PTS

Current General Dynamics' F-16 Training Devices incorporate the new generation of processor technology which supports Design area aerodynamic and flight controls software. Reuse of this software will permit much more rapid devleopment of the devices, reduce the validation requirements for the flowdown moudles, and permit the PTS design to remain concurrent with the aircraft. IPTs are coordinating the data requirements and development plans to ensure smooth flowdown to the PTS.

NEW PROGRAMS

New aircraft programs at General Dynamics are incorporating the new technology processors into the EFS systems, permitting reuse of the design area software in the EFS. Software Development Plans (SDP) developed for the EFS will reflect the level of formality required by the PTS, and PTS programs will incorporate reuse of this software.

The planning for this level of interaction between functional areas is enabled by the use of IPTs. For example, IPTs which manage the flight control and pilot training systems are working together to insure an integrated plan for design and development of both the control systems, EFS and PTS.

This approach supports the incorporation of requirements from the PTS area into the EFS, which in turn permits the EFS design to accomodate requirements for failure modes, control logic and other functions required by the PTS. In addition to incorporating PTS requirements into the EFS, this process also incorporates the PTS function itself into the EFS function for design and development of EFS software.

In these programs, a "crossflow" team is formed, consisting of engineers from the EFS and PTS areas. Although the EFS is developed under the management of the Engineering Simulation function, integrated participation of the PTS function is assured in development of the overall plan, detailed development of the Software Requirements Specification (SRS), Software Development Plan (SDP), all design reviews, software development, test and integration.

Development of this software in common at the beginning provides a strong baseline for continuing updates as the aircraft design evolves, permitting ongoing updates to the EFS to smoothly integrate into the PTS. Figure 3 shows the integrated development cycle for EFS and PTS with flowdown implemented.

CONCLUSIONS

While the history of development of flight simulators for Engineering and Training applications has shown a tendency for divergence which results in lengthy development schedules and high costs, there have been some examples of software reuse which have mitigated this problem to some extent. These examples, however, have been by and large "synthesis after the fact". With the advent of fast, inexpensive and flexible microprocessor-based computer systems and the "cultural paradigm shifts" manifested in improved software processes and Integrated Product Teams, a synergism of three phases of the aircraft design process (functional area design, engineering simulation and pilot training simulation) now seems feasible.

This synergism manifests itself primarily in the reuse of software among the three phases, but the significance is much deeper. Properly done, this synthesis incorporates wholistic integrated team planning sufficiently early in the design program to ensure early requirements definition and to support the transportation and management of design data and documentation from one phase of the design cycle to the next. The integrated process resulting from these technical and cultural innovations removes traditional barriers from the design process, and reduces associated schedule and cost elements from new military aircraft programs.

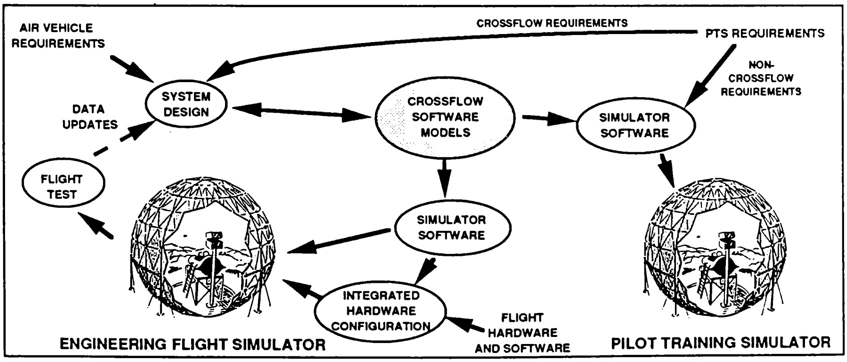


FIGURE 3. DEVELOPMENT OF EFS AND PTS SYSTEMS SHOWING CROSSFLOW SOFTWARE

REFERENCES

1. Baldwin, D., Drewett, J. D., "Air Combat Simulator and its Role in the Aircraft Development Process", AIAA 862682, October1986

2. "Simulator Data Integrity Study", ASD-TR-88-5024, April 1988

3. Haseltine, E. , "Trainer Concurrency- Problems and Solutions", Proceedings of the 1989 I/ITSC, November, 1989

ACKNOWLEDGEMENTS

The authors would like to thank Mr. Roger Marr, Bell Helicopter Textron, and Mr. Phillip Graef, McDonnell Douglas Training Systems, for examples of "flowdown" in their companies.

AIAA-92-4162-CP

MANIPULATION AND MANAGEMENT OF DATA COLLECTED AT THE CREW STATION RESEARCH AND DEVELOPMENT FACILITY: A CASE STUDY

Steven B. Rogers
Crew Station R&D Branch
Moffett Field, CA

David Kennedy
Monterey Technologies Inc.
Carmel, CA

Perry Meade
CAE Electronics, Ltd.
St. Laurent, Quebec, Canada

Abstract

This paper discusses issues relevant to the collection, manipulation, and storage of full combat mission flight simulation data. Specific examples are drawn from the data reduction and analysis system developed for use at the U.S. Army's Crew Station Research and Development Facility (CSRDF). The system's current capabilities, as well as plans for its future improvement are addressed.

Introduction

The high sampling rates frequently used to gather simulation data often leave the researcher with an overwhelming amount of data from which to draw conclusions about an experiment. Often, so much is collected that the data become unwieldy and difficult to manipulate following the simulation.

In seeking to address issues relevant to the manipulation and management of simulation data, the following paper will provide examples from the data collection and reduction system used at the U.S. Army's Crew Station Research and Development Facility (CSRDF). Examining the specifics of this system will hopefully yield insight into factors which have impact on this process. Although the information offered here is in many ways specific to the CSRDF, it is hoped that many of the issues will be applicable to other systems as well.

The Crew Station Research and Development Facility

Overview. The CSRDF is a state-of-the-art rotorcraft simulator designed to address human factors issues in a simulated combat environment. The facility permits man-in-the-loop simulations of part- and full-mission military operations in an interactive environment that can include up to 11 friendly or hostile aircraft and 100 ground threats.

Run durations vary depending on the type of experiment being conducted, but a full mission simulation including ingress and egress may last up to 1½ hours.

File sizes vary a great deal from simulation to simulation, but average sizes are on the order of 25 Mb and have been as large as 60 Mb. A single experiment might require as many as 60 such files.

<u>Data Flow.</u> The flow of data from time of collection until delivery to the researcher is depicted in Figure 1.

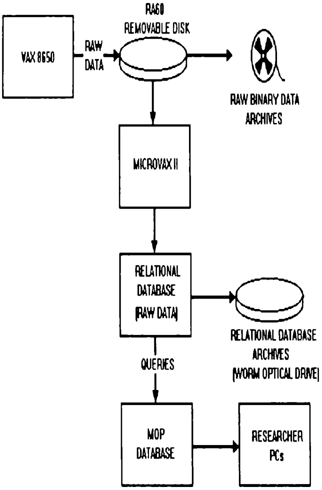


Figure 1: CSRDF Data Flow

Data collection is performed by the simulator's host computer, a VAX 8650. These data are written to an RA60 removable hard disk. Following each day's collection, the raw data are backed up to nine-track tape and the RA60 collection disk is removed to a MicroVAX II. Here the raw data are loaded into a relational database and then archived to a WORM (write once, read many) optical disk.

A database is created for each data run. Once the data are in the database, they are queried by programs that combine a high level language (C) with the database's data manipulation statements to extract measures of performance (MOPs). The extracted measures for each run are output to a single MOP database that holds the reduced data for the entire experiment. This database can then be queried to produce ASCII files which may be downloaded to PCs for analysis by the researchers.

<u>Data Considerations During Planning, Collection and Processing of the Data</u>

Data manipulation and management topics fall naturally into three categories: planning, collection, and processing. These categories correspond to the pre-simulation, run-time, and post-simulation phases of the simulation, respectively.

Providing an important backdrop to the data collection process are three pieces of information: 1) the goals and hypotheses supporting the simulation, 2) the experimental design being used, and most importantly 3) the high level performance measures of interest in the simulation. Although it is not critical that any of these items be made known to the data team, it is a better guarantee that the correct data will be collected if they are.

Planning. What is done in this phase greatly influences the way data reduction and analysis procedures are carried out. It is during this phase that decisions are made as to the methods which will be used to provide the end results to the researchers.

Collection Strategies. Important to both manipulation and storage of the data is the question of how the data will be collected.

Both event-driven and continuously-sampled (periodic) data can be collected at the CSRDF. Either type can be disabled from the control console at run-time. Most simulations collect both types of data.

"Events" are used to log discrete occurrences of interest during a simulation. Examples include weapon launches, sensor detections by threat, etc. Each event log contains the name of the event, the time it occurred, and up to six associated parameters, such as the aircraft's altitude at the time of the event. Presently, the CSRDF has the ability to log any of 400 events.

Periodic recording lists consist of variables selected from a pool of several thousand variables available from a common database at run-time. These data can be sampled at 15, 30, or 60 Hz, with the possibility of collecting at any or all of these rates within a single simulation run. For a typical CSRDF simulation, most of the data are collected at 15 Hz, with a smaller number of variables collected at 30 Hz, and fewer still at 60 Hz. There will almost always be a need to collect some periodic data.

Ideally, much of the data needed to construct performance measures could be collected as events. Data would then be written only when the occurrence criteria for each event had been satisfied. This would result in considerable storage savings for events which seldom, if ever, occur.

On some simulators it may be relatively difficult to create new events or modify old ones. At the CSRDF, events are logged through routine calls in the real-time software. To create or modify events requires familiarity with and modification to the simulation software.

Future improvements to the CSRDF's data collection system will include transition to a completely event-based system. The capability to collect periodic data will still exist by using events whose only criterion for occurrence will be the amount of time elapsed since they were last logged. Included in the upgraded system will be an event-editor that will allow relatively easy creation and modification of events, while requiring little special knowledge on the part of the user.

Sampling Rates. For data which must be sampled frequently, care should be taken that individual variables are not sampled more frequently than their update rate in the simulation. This will help avoid redundancy in the data and help save storage space.

Careful consideration should be given to sampling variables at a rate slower than their simulation update rate. This is especially true for variables being used to indicate momentary state changes. There is a danger that these changes may be missed at the slower rate.

Ideally, one would have a database which could be dedicated to keeping up with simulation metadata such as variable update rates, data types, etc. In an automated system, the final data collection list could be checked against this database to ensure data redundancy is avoided.

Data Marking. Not all of the events of interest in a simulation can be collected automatically. Nor is it always possible to anticipate what these will be prior to simulation. Detecting and logging these events may require an observer watching the simulation.

Data marking refers to the mechanism by which markers are entered into the data stream by an observer, either to indicate the occurrence of single discrete events or to segment the data into periods of interest. The marker itself may be nothing more than a variable whose value increments every time the observer presses a button. During data manipulation these markers can be used by database queries to quickly locate relevant segments of data. If data markers are to be used to construct reliable measures of performance, then clear event definitions must be established during the planning stages of the experiment.

There are several reasons why data marking should be used sparingly. Data marks are not as precise or as reliable as automatically collected data. Because they rely on human input, they are prone to false alarms and misses, whereas automatic forms of collection are generally not. A second reason is the burden that data marking places on the data logger. Data marking requires a high level of attentiveness that is difficult to maintain over the course of a run. Even if a small number of events are involved, confusion may result if many of the events are likely to occur within a short period of time.

Depending upon the provisions made for data marking, there may also be some overhead associated with post-processing of the markers. At the CSRDF, a handwritten record of marker values and the events they mark must be kept at run-time so that their meaning can be determined following the run. Making this information accessible to programs requires creation of external files detailing this correspondence.

An improved system would circumvent the need to create these external files by providing a mechanism for logging marker events directly into the data stream. This would be feasible if there were a number of predefined marker events of interest in the simulation. The event entry would include an event name, a short description of the event, and the time the marker button was pressed. The system's real-time interface might consist of a set of programmable buttons (provided through hardware or software) that could be linked to a set of event definitions for each simulation.

Recording Verification. Sample data should be gathered during a dry run in which all planned procedures are exactly followed. In particular, the length of the run and the data that are collected should be representative of what is expected in the actual simulation.

The sample data should be used to:

1) *Verify data list collection.* Variables (particularly temporary ones) sometimes change names and/or function between simulations. All variables on the data list should, therefore, be checked to see that collection is in fact taking place and that values are within expected ranges.

2) *Identify additional collection needs.* For a particular part-task simulation, the CSRDF's data team was surprised to discover weapon trigger pulls were being missed because they had failed to record the left trigger state (not all pilots are right-handed). These kinds of mistakes can be averted by careful examination of dry run data. It also emphasizes the need to collect such data under varied conditions and with different test subjects.

3) *Verify available recording time.* Recording time must be sufficient to allow runs of the projected duration to take place. The recording time will depend on the number of variables being recorded, their sampling rates, and the amount of space available on the recording media. If the available recording time is less than the projected run length, then adjustments will have to be made to either the recording list size or data resolution.

Collection. The number of factors which apply to the actual collection of data and affect efficient data management and manipulation are few when compared to those involved in planning.

Data Recording Control. The data recording system should provide the ability to freeze data collection. Whether or not this will be appropriate for an individual simulation will depend on the goals and conduct of that simulation. For those instances where it is appropriate, the result will be a savings in file size.

The run-time interface to data collection should consist of controls which are both quickly accessible and easy to use. If several types of recording media are being used for collection (e.g. audio, video), the interface should provide a means to control those systems both separately and jointly.

As mentioned earlier, the CSRDF provides separate and joint control of periodic and event-driven data collection. These sources may also be coordinated with the facility's collection of audio and video data.

Simulation Aborts and Freezes. Few things are as frustrating as to be near the completion of an 1½ hour run only to have the data lost because of a simulator crash. Occasionally simulators abort for whatever reason, and when they do, there should be a mechanism in place to handle the data files which may be left open in the process. Data loss due to crashes occur primarily in simulators which use hard drives for data collection. Simulators which use tape as the collection media are not as prone to this problem.

Run Documentation. There are essentially two types of documentation which play an important role in later processing and analysis of the data: 1) run-time documentation, and 2) experimental run information.

Run-time documentation consists of comments made either by the experimenter, the simulation operator, or the pilot during the run. These comments may prove invaluable later when trying to explain anomalous data. Without such comments, much time can be wasted trying to reconstruct what happened during the run.

Documentation of the experimental run information becomes important during statistical analysis of the data. This information pertains to the exact experimental conditions which characterize individual runs. During analysis these are treated as levels of the independent variables and blocking variables in the experimental design. Once the performance measures for the entire experiment have been constructed, it becomes necessary to manipulate and group the data from the various runs according to the experimental conditions under which they were collected.

At the CSRDF, this is accomplished by entering the experimental run information for each run into the experiment's MOP database and then using the database's data manipulation language to manipulate the data accordingly. Currently, this requires creation of an external file to hold the information. This file must then be read by a program which loads the information into the MOP database. Since every simulation is different, this program must be tailored to the needs of each experiment.

Improvements to this process call for the development of a run-time utility which will allow the user to enter the information directly into the data files just prior to the beginning of each run. Although it is currently possible to enter this information into the data file, the system allows such free form input that the inconsistencies preclude being able to query them later. The new utility will greatly reduce the number of inconsistencies and errors that occur during the data entry process.

Processing. Once the data have been collected, they must undergo additional processing to create the higher level measures of performance. Construction of these measures can be a time-consuming process. All aspects of the data's treatment following an experiment, including how the data should be formatted and where they should be stored, need to be geared toward shortening this process.

Databases as Intermediate Data Repositories. Storing the data in a relational database format offers several advantages. Depending on the database product being used these might include:

1) *Data Organization.* Data can be logically grouped within the database. Data which belong together can be placed together within database tables.

A more subtle form of data organization comes in the form of index creation. These are invaluable when it comes to increasing the speed with which the data can be accessed and retrieved.

2) *Data Manipulation.* Databases offer ways to both explicitly and implicitly manipulate the data.

Explicit manipulation of the data is most likely to occur through the database's data manipulation language (DML). For most databases the DML provides a powerful interface to the data. This includes the ability to extract subsets of the data (variables may be accessed by name, and rows by criteria that are established by the user), the ability to join data from different tables (e.g. event data with periodic data), and the ability to sort the data in some fashion.

The database may also provide ways to *implicitly* manipulate the data. The database used at the CSRDF allows column (variable) values to be based upon computational combinations of other variables in the same row of a table. These computations do not have to be calculated by the user, but are accomplished by the database itself and are transparent to the user.

3) *Data Validation.* Constraints may be able to be placed on database columns so that implicit checking of the data can occur when the data are stored.

4) *Configuration Management.* The database product used at the CSRDF allows data definitions to be stored in a data dictionary. These definitions can then be inherited by other programs. This removes the burden of ensuring data type compatibility from the programmer.

5) *Database Utilities.* A number of utilities may be provided along with the database. At the CSRDF, these include a utility for interactive database queries and precompilers that allow database statements to be integrated into high level C language programs.

<u>Data Visualization.</u> The ability to view a graphic or video playback of what occurred during a run is an indispensable tool to data processing. The alternative approach of watching endless streams of data scroll by on the computer screen can result in hours of wasted time spent investigating anomalous data.

If a graphics playback is used, it may even be possible to use it as a debugging tool for code development. At the CSRDF, database queries have been modified to produce graphic displays which not only show how a particular run progressed, but also what decisions were being made about the run by the program logic. This allowed the algorithm in question to be developed much more quickly than would have been possible otherwise.

A high degree of fidelity, although desirable, may not be needed in order for the playback to be effective. At the CSRDF, a simple x,y plot of threat and friendly positions is often all that is needed for code development and debugging.

If a high fidelity playback *is* available, however, it opens up another interesting possibility. Consider the situation where some of the data could <u>not</u> be collected during real-time because of the processing limitations of the simulator. A possible solution might be not to record the data during the run, but instead to collect it <u>after</u> the run *during the playback itself.* This would **<u>only</u>** be possible, of course, if the same algorithms and software, which were used in the actual run could again be used to generate the data during the playback.

Plans are underway to provide an interactive playback facility at CSRDF. Advantages of such a system are to provide "what if?" types of modifications to the data collected at run-time. Also, the post-simulation observer will have the ability to add data markers to the data to isolate times in the data where items of interest have occurred.

<u>Data Delivery.</u> Due to the number of research groups at

the CSRDF the versatility of delivery of final-form data is vital. The CSRDF has the ability to deliver data in many formats on most forms of media.

The data formats supported at the CSRDF include database formats (Rdb, RBR, dBASE), standard ASCII, binary, and Data Interchange Format (DIF). The use of custom C language routines can accommodate a virtually limitless array of formats which may be needed to transfer data between hardware platforms.

Transfer media must also be as diverse as the equipment used by the researchers. At the CSRDF Ethernet and serial data communication lines are routinely used for direct data transfer to local researchers. For systems which are not linked electronically to our network both sequential and direct access media are available.

Sequential access media supported at the CSRDF include 4mm, 8mm, 9 track and TK50 tape. The advantage of using this form of transfer is that it is a fairly durable, inexpensive, and dense medium. For example, a single 8mm tape which costs under $10 can hold up to 5 Gb of data.

Direct access media used at the CSRDF include several types of removable hard drives including Bernoulli, DEC RA60, and SCSI drives. However, the relative expense of this form of media is often prohibitive for mass transfer of data. Also supported are floppy diskettes for delivery of smaller files.

In some instances graphic representations of the data can be provided either on paper or electronically.

Summary

At present, the CSRDF is undergoing a major upgrade which will incorporate many of the improvements mentioned in this paper. It is expected that these modifications will enhance the data collection capabilities of the CSRDF and expand the services which may be offered to our researchers.

The task of turning raw data into useable performance measures usually requires many "behind the scenes" steps and considerations which are not always obvious. It is hoped that this paper has served to shed light on some of these considerations so that researchers at other sites may benefit from our experiences.

It should be noted that the views reflected in this paper are those of the authors only and are not necessarily the views of any Governmental agency. References to specific hardware and software are for the readers information only and do not constitute an endorsement of these products.

AIAA-92-4164-CP

APPLYING DIGITAL MUSIC INSTRUMENT TECHNOLOGY TO AIR-COMBAT SIMULATION

Mark E. Sturgell*
Electrical Engineer
WL/FIGD
WPAFB, OH 45433

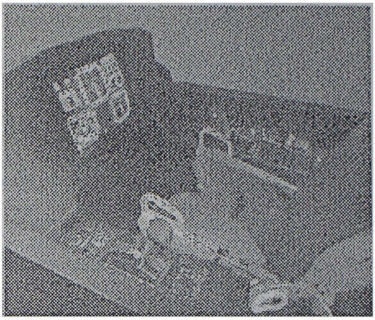
2d Lt Jeffrey M. Hebert
Electrical Engineer
WL/FIGD
WPAFB, OH 45433

Abstract

Recent advances in digital music instrument technology have dramatically changed the way in which musicians create and produce music. With the digitally based music synthesizers available on the commercial market, it is possible to re-create virtually any sound with an unprecedented level of realism. At the Wright Laboratory (WL) Engineering Flight Simulation Facility, this technology has been borrowed from the professional music industry to produce an extremely realistic and low cost sound effects capability for piloted air-combat simulation.

In addition to the decreased engineering costs associated with using commercial off-the-shelf equipment, taking this particular approach to generating simulator sound cues has produced many other benefits. In earlier simulation sound systems, computers had to generate numerous discrete and analog signals to control sound generating hardware in real-time. Most of today's music synthesizers contain specialized digital signal processors that can generate extremely complex sounds in real-time. Many also use microprocessor front-ends to provide control over these sounds from an external host computer. In the context of flight simulators this distributed processing translates into less computational overhead for the simulation host computer.

The advantages of external control in live performance have prompted the music industry to adopt a Musical Instrument Digital Interface (MIDI) standard[1]. Spearheaded by the International MIDI Association (IMA), MIDI has gained wide acceptance across the electronic musical instrument industry as the de-facto standard for performance control of electronic instruments. Being specialized for control of musical expression, it can afford the simulation programmer the same subtle control over the generation of sound effects as a musician has over the nuances of musical notes. And since it is a real-time control protocol, no appreciable transport delay would be incurred.



Choosing digital sampling synthesizers to generate cockpit sound effects has provided the benefit of added realism. Unlike other synthesizer architectures which generate complex sounds by modulating and combining simple periodic waveforms, samplers start with digitized recordings of real-world sounds. Sophisticated built-in software allows precise control over the shape and contour of the sound. Designing sounds this way is extremely easy and the results are often stunningly realistic.

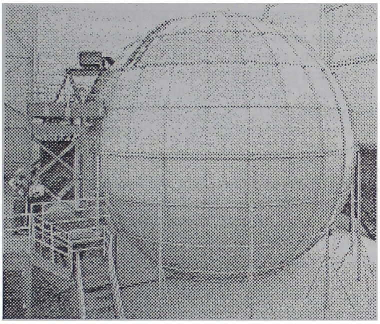
While each of these advantages are significant, the most important argument for using sampler based sound cueing systems is the realism of the result. Taken together with good visual and motion cues, a comprehensive sonic environment increases the ability to suspend disbelief in piloted air-combat simulations. This in turn helps yield more accurate simulation test results.

This paper discusses in detail the development and implementation of a digital sampling synthesizer based sound effects system for piloted combat mission simulations at the WL Engineering Flight Simulation Facility. Beginning with the initial simulation requirements and continuing through to results obtained in actual piloted combat simulations this paper will highlight the simplicity and flexibility of the systems as well as the effectiveness this exploitation of commercial technology has provided the Air Force.

*Member, AIAA

WL Engineering Flight Simulation Facility

The WL Engineering Flight Simulation Facility is the preeminent Air Force facility for assessing and integrating advanced aerospace technologies through simulation. The facility is operated and maintained by the Control Integration and Assessment Branch, Flight Control Division of Wright Laboratory's Flight Dynamics Directorate. While traditionally the branch has analyzed and tested flight control systems, today's efforts focus more on what is called, "control of flight" rather than, "flight control".



Mission Simulator - 1

Toward that end, the branch has re-oriented the facility to support a Tactical Mission Simulation (TMS) capability. At the heart of this new capability stands the Mission Simulator One (MS-1), a 40-ft, 360°-field-of-view spherical simulator. The generic cockpit at it's center is capable of emulating nearly any modern fighter aircraft. Supporting the TMS are four simplified Manned Combat Station (MCS) cockpits. These cockpits allow more pilots to participate in air-battle simulation scenarios without the expense of constructing and maintaining additional Mission Simulators. Completing the TMS simulator cockpit suite is the Large Amplitude Multi-mode Aerospace Research Simulator (LAMARS). The LAMARS also plays a vital role in supporting flying qualities studies.

Several Gould/Encore simulation host computers, Silicon Graphics workstations and a CompuScene IV-A image generator provide the computational power needed to drive the cockpits in realistic combat mission scenarios. I/O is distributed using a Computer Automated Measurement And Control (CAMAC) fiber-optic network.

Addressing A Need

In 1988, the Flight Dynamics Directorate (FDD) began an advanced technology transition program called Integrated Control and Avionics for Air Superiority (ICAAS) This program would use the TMS facility to assess changes in pilot effectiveness with new air-combat algorithms and displays. The requirement to augment the TMS environment with audio cues and the unique solution to that requirement is the focus of this paper.

Arguably, audio cues in and of themselves are not a predominant factor, but taken together with visual cues significantly increase a pilot's ability to become mentally immersed into an air-combat scenario. Since ICAAS focuses on assessing the effectiveness of new tactics algorithms, this ability would have a significant impact on test results.

The candidate sound effects system should support basic environmental effects like engine noise and machine gun fire as well as specific avionics cues for the ADV-166 (the ICAAS F-15 aircraft). In addition, it should be adaptable to the sound effects requirements of a wide range of fighter aircraft in order to remain useful for other aircraft simulations that may be conducted at the WL Engineering Flight Simulation Facility. Initially the sound effects system would have to handle six Man-In-The-Loop cockpits yet be expandable to twelve to accommodate future ICAAS experiments. The system must integrate readily into the simulation facility within ICAAS schedule constraints so off-the-shelf technology should be used wherever possible. Finally, costs should be kept down to fit within budget.

Looking across the spectrum of possibilities, several potential solutions present themselves. The most obvious choice is to purchase a traditional simulation sound system from aerospace contracting company. These systems are designed to integrate with aircraft simulators and would require little effort to implement. However, they are generally implemented only in hardware which makes them inflexible. Newer software-driven designs are also available but their cost is quite high. Another consideration is how closely these turn-key systems can fit into the unique requirements for the simulation facility.

An alternative would be to design a customized sound system from scratch. Such a system could be tailored to fit all ICAAS requirements without compromise. While this approach provides the most flexibility, it carries some heavy penalties. In addition to being the most expensive in terms of direct engineering costs, it is unlikely that a system designed completely from scratch could be implemented in time to meet the ICAAS schedule.

A good compromise would be to use off-the-shelf subsystem components and customize the overall system to suit the facility. With a little creative engineering, digital sampling music synthesizers could be employed to produce a remarkably flexible, low cost sound system. The technology developed to provide musicians with real-time control over sound could be used to efficiently control aircraft sound effects as well. Normally, the work conducted at Wright Laboratory is transferred outward to the commercial sector. In this instance, an opportunity to transfer technology *into* the Air Force allowed us to leverage the cost/benefit ratio of the sound system.

While this solution afforded the most reasonable tradeoff between cost and function, there was some initial concern about the integration of musical instrument technology into an air-combat simulator. If not for the outstanding interoperability between synthesizers and simulation computers afforded by a MIDI control interface, integration costs may have been prohibitively high.

Selecting a particular sampler required surveying available units. High-end audio workstations like the E-Mu, New England Digital Sinclavier, and Fairlight CMI proved to be too expensive. These production-quality systems were simply more than was needed. Lower cost samplers including the Ensoniq Mirage, Akai S-1000 and Ensoniq Performance Sampler (EPS) would have been more in line with the requirements. The Mirage was eliminated due to it's 8-bit resolution and inadequate support. Of the remaining two, the Akai S-1000, while superior to the EPS in audio quality did not provide as flexible a MIDI implementation. In fact, the overriding factor in EPS's favor was its robust MIDI control implementation. All considered, the EPS proved to be the sampler of choice. Since a musical keyboard was not required, the rack mountable version , the EPS-m, was purchased.

Sound System Development

Having selected the sampling synthesizer platform, the next step was to devise a suitable scheme for developing sound sets and controlling them in real-time. In parallel with this effort, appropriate sound reinforcement gear had to be acquired and integrated into the facility.

Sound Effects Development

For ICAAS, the choice of sound effects are specified directly by simulation requirements documents. The specification subdivides these sounds into two major categories: environmental sounds which represent the mechano-acoustical environment, and avionics tones and alarms presented to the pilot to aid in situational awareness. This subdivision turns out to be a natural one which can be employed by any aircraft simulation.

Developing a standard scheme for implementing sets of sound effects was guided by the EPS architecture. The EPS regards collections of samples over it's 128 note range as an *Instrument.*[2] For simulation purposes this grouping is referred to as a sound-set. This logical subdivision makes it easy to associate a collection of sound effects with a particular aircraft. Independent sound sets can be developed from libraries of individual effects and pieces of other sound sets.

Creating a sound set for flight simulation requires some ingenuity and resourcefulness on the part of the engineer. Audio tape recordings, sampling live sound or employing algorithmic synthesis are the most popular means to create simulation sound effects.

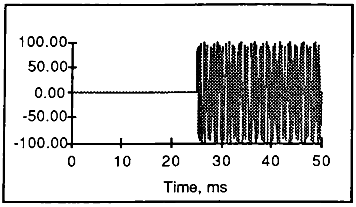
Audio tape recordings are the easiest media to work with when sampling. Interfacing a tape player to a sampler is straightforward since most tape players provide audio outputs compatible with the line level inputs on a sampler. Furthermore, during the sound development process, tape provides the opportunity experiment with pre-equalization or other treatments to the tape output before sampling the sound.

Sometimes the best way to get a realistic sample is to literally place microphones at a pilot's ear points and make recordings of everything he or she would hear during the phases of flight of interest. Some sounds such as aerodynamic noise would require the aircraft to be airborne and in a stable attitude. Other tones, such as avionics tones and warnings, can be recorded on the ground, or even on a lab bench.

One problem with sampling in the real world is isolating the target sound from the background sounds. For example, when trying to sample an avionics tone from an in-flight recording there may be unwanted sounds from the aircraft electrical system, engine noise, or inopportune pilot chatter that clutters the sample. Digitally extracting a single element from a sample like this is extremely difficult.

A better alternative is to create tones and alarms from a mathematical specification. For example, the

Critical Angle of Attack (AOA) tone used for ICAAS consists of a 1600 Hz sine tone pulsed on and off at a 20 Hz rate.

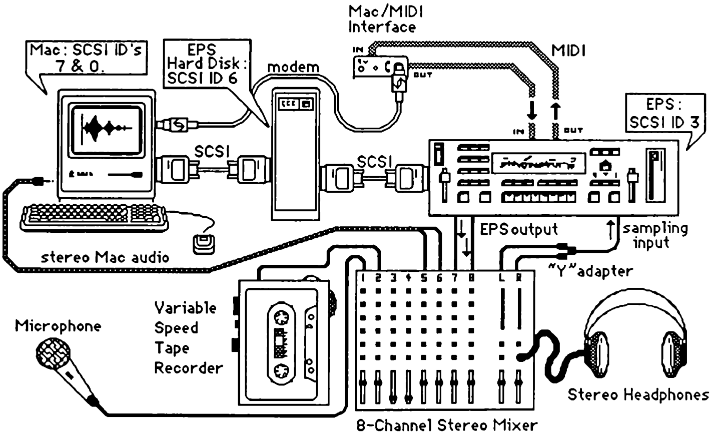


Critical Angle-of-Attack Tone

Assuming a 50% duty cycle, this means the tone would be present half the period of 20 Hz, or 25 ms. This waveform, played repetitively from end to end will produce the desired tone.

Most often it is not economically feasible or physically possible to record a cockpit sound live. Sometimes improvisation is called for. A colleague at NASA Ames[4] related a story about his attempt to produce the sound of a missile impacting the airframe. Sampling this phenomenon live obviously has its drawbacks. Instead, a microphone was placed inside a garbage dumpster, and with a baseball bat striking the side of the dumpster he created the missile impact sound. Some additional cleanup and treatment of the sound was required, but in the end, the sample was a convincing cue.

Once a sound has been sampled it usually requires some form of treatment. A sound effects workstation has been developed to prepare sounds for a simulation. As shown below, the workstation consists of a Macintosh computer, an EPS, a SCSI hard disk, an audio mixer, a variable speed tape recorder, a microphone and stereo headset.



Sound System Development Workstation

In this setup, audio to be sampled is fed through the mixer to the EPS sampling input from either the tape recorder or the microphone. The EPS performs the digitization of the audio and can send the digital audio to the Macintosh for editing and refinement. Specialized audio editing software runs on the Macintosh for this purpose.

When some sounds such as the AOA tone discussed earlier are played back by a sampler, they are looped continuously. In other words, the AOA sample itself lasts only 50 ms, but it is repeated hundreds of times achieve the desired length. Care must be taken to assure that the beginning and end of the loop match up to form a continuous wave shape. Otherwise the loop will pop or click each time it reaches the splice point. For this reason, the best splice point for the beginning and end of a sample are those which occur at the zero crossing, and continue in the same direction. The software on the Macintosh helps to locate precise crosspoints for the loop start and end.

Crossfading multiple samples together into a single sound effect is a very powerful technique for handling complex dynamics. For example, the ICAAS engine noise effect illustrates how crossfading two samples improves realism. One sample contains the pitched component of the engine sound and represents the turbine whine. The other sample contains the white noise component and represents fuel combustion and engine thrust. The relative amplitude, pitch and frequency content of both samples are modulated in real-time in response to the throttle position. At idle, the pitched component dominates the white noise component. As the throttle is increased, the pitched component increases in frequency yet decreases in amplitude relative to the white noise component. At full mil-throttle, the white noise sample dominates the effect.

The main advantage of having simulator sounds in software instead of hardware or firmware is the ability to rapidly reconfigure a sound set. During the development of the ICAAS sound set, pilot feedback about the sounds was incorporated and turned around in minutes. For example, one pilot commented that the engine afterburner sound didn't have enough "grumble". While it is difficult to quantify what *grumble* actually is, the solution was to lower the root key of the afterburner sound by two octaves while also bringing a bandpass filter's center frequency downward to accommodate the pitch change and remove the higher frequency harmonics. The new sound was incorporated into the sound set and was ready for the next simulation run.

If necessary the cycle can be repeated by taking pilot feedback, determining a course of action to correct the sound, and re-incorporating it into the sound set. For even faster turn around, placing pilots directly in the sound editing cycle with the engineer allows them to tune filter values and amplitudes at the sound development workstation, until they say, "Now that's grumble!"

Sound Effects Control

The majority of MIDI communication is based upon two or three byte messages consisting of one Status byte and one or two Data bytes. Status bytes are used to determine the message type and number of Data bytes that follow it. MIDI messages are used to control every aspect of sound generation.

```
STATUS    DATA
1001nnnn | 0kkkkkkk 0vvvvvvv

nnnn: channel number (1-16)
kkkkkkk: note number (0-127)
vvvvvvv: ≠ 0: velocity
vvvvvvv: = 0: Note Off
```

Note On: A typical MIDI Message

Each of the 26 sound effects in the ICAAS sound-set was created or sampled and then assigned a specific note number. Thus if the sound-set shown below was loaded onto an EPS with a keyboard and Middle C (C4) was depressed, you would hear the sampled voice warning, "Warning! Warning!".

Note	Hex	Wavesample Name
C2	24	Left Engine, Afterburner
C#2	25	Right Engine, Afterburner
D2	26	Left Engine, Canopy Closed
D#2	27	Right Engine, Canopy Closed
E2	28	Aerodynamics, Clean
F#2	2A	Missile Launch, Right Side
G2	2B	Missile Launch, Left Side
G#2	2C	Landing Gear Locking
A2	2D	Gun - 4000 RPM
A#2	2E	Gun - 6000 RPM
D#3	33	Over-G Warning Tone
E3	34	AIM-9 Missile Tone
F3	35	AIM-9 Missile Lock-on
F#3	36	IFF Mode-4 Tone
G3	37	TEWS Caution
G#3	38	TEWS Launch
A3	39	AOA Tone
A#3	3A	J/P3 Missile Tone
B3	3B	J/P3 Missile Lock-on
C4	3C	"Warning! Warning!"
C#4	3D	"Warning! Fuel Low!"
D4	3E	"Over-G! Over-G!"
D#4	3F	"Bingo Fuel!"
E4	40	"Warning Engine Fire - Right!"
F4	41	"Warning Engine Fire - Left!"

ICAAS Note Assignments for MS-1

While a sampled sound is an accurate representation of sound at a specific point in time, many sounds change their pitch, amplitude and tone over time. A jet turbofan engine, for example, makes a sound which varies in amplitude and pitch as a function of the throttle position.

Any dynamic aspect of a sound effect can be modulated using an *aftertouch* message. There are two types of aftertouch messages: Channel Pressure and Polyphonic Key Pressure. Channel Pressure effects all the notes on an instrument simultaneously. Polyphonic Key Pressure, on the other hand, affects notes individually, which makes it an ideal modulator for sound effects sets.

Driver software written in FORTRAN/77+ has been developed to allow simulation engineers to easily add sound effects to the simulation executive. Pre-declared variables in header files allow sounds to be addressed by name rather than by note number. The following table lists a subset of the functions implemented in the driver.

Function Name	**Purpose**
SoundOn(cockpit, note)	Turns on a specified sound (note) a specified cockpit.
SoundOff(cockpit, note)	Turns off a specified sound (note) a specified cockpit.
SoundOnOff(cockpit, note)	Turns on a specified sound (note) on, then off, at a specified cockpit. This is useful for discrete events.
Toggle(noteon, cockpit, note)	Toggles on or off a specified sound (note) at a specified cockpit given the current state of the sound (noteon).
Cockpit_Sounds_Off(cockpit)	Turns off all sounds at a specified cockpit. Useful as a kill switch.
Modulate(min, max, value, lastmod, noteon, cockpit, note)	Modulate a specified sound (note) between given minimum and maximum values. Wavesample should have pitch, filter and and amplitude parameters set to be modulated by polyphonic aftertouch messages

Sound System Driver Routines

At this time there are no commercially available MIDI interface cards for the CAMAC bus. However, there are many RS-232 cards available. The highest data rate supported by the standard CAMAC serial card is 19.2 Kbps. This creates the potential for an I/O bottleneck. As an interim measure, an RS-232 to MIDI converter was developed. The converter simply takes an RS-232 signal at 19.2 Kbps, converts it to a 5 mA current loop signal at the standard MIDI rate of 31.25 Kbps. This converter generates only MIDI OUT messages; it cannot receive MIDI IN messages.

Although MIDI data is asynchronous in nature, MIDI commands are synchronized to the simulation frame rate. Therefore, assuming a three byte MIDI message length, given a limit of 20 simultaneous voices and an effective data rate of 19.2 Kbps, the highest rate at which a MIDI output routine can be scheduled is:

$$Rate = \frac{19200\ bps}{20\ voices \times 3\ bytes \times 8\ bits/byte} = 40\ Hz$$

Is this a problem? Manzanowski[3] related the ability of human hearing to various types of frequency and amplitude changes at different update rates. He suggests that 30 Hz is typically the maximum update rate for sufficient detection of pitch changes. His results are summarized in the following table:

Type of Signal	**Update Rate**
Periodic waveform pitch changes	30 Hz
Random signal filter cutoff frequency changes	20 Hz
Periodic waveform amplitude changes	10 Hz
Random waveform amplitude changes	10 Hz
Modulation level changes	10 Hz
Stationary signal on/off commands	10 Hz

Audio Reinforcement

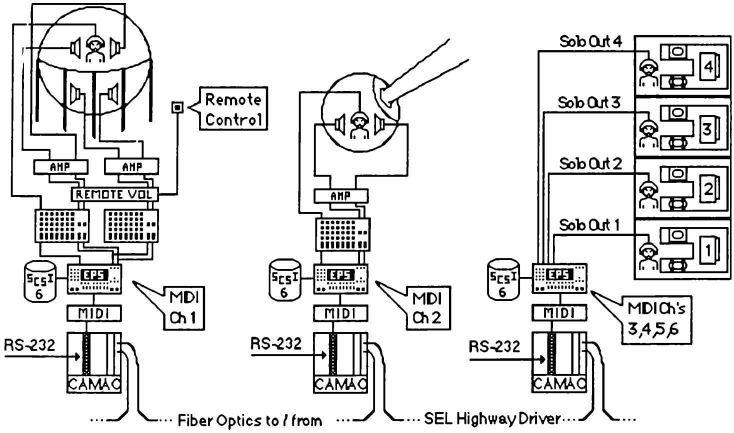
The specification of audio reinforcement and distribution gear for the sound system was driven primarily by ICAAS requirements. However, the physical dimensions of the facility and acoustic properties of the simulator cockpits had to be considered as well.

Yamaha P2075 audio amplifiers and small triaxial speakers were chosen to provide the basic audio reinforcement for MS-1 and LAMARS. Pairs of speakers were mounted toward the rear of each cockpit and aimed directly toward the pilot's head position. This speaker placement insured that the pilots would hear predominantly near-field sound, thus reducing the annoyance caused by multiple reverberations within the spherical simulator space. A more powerful Yamaha P2150 amplifier and pair of 15-inch studio monitor speakers were used to produce the desired vibrational cues for the MS-1. The studio monitors were rigidly mounted beneath the cockpit pier using specially designed brackets that allowed adjustment in elevation and azimuth. This way, low-frequency vibrations could be transmitted to the pilot by taking advantage of resonances within the 40-foot sphere as well as through physical contact.

Since each cockpit required it's own sound effects channel, audio mixers provided the most flexible means of controlling individual volume and tone. Three low cost mixers, each with eight inputs, two main line-level outputs and two effects loops adequately handled all of the sound effects signal routing. Two of the mixers were dedicated to the MS-1. The first accepted a left and right channel stereo image from the EPS main outputs, boosted the low frequencies, and sent them to the P2150 to provide vibration cues. The second also accepted a stereo pair from the EPS, but did not alter the frequency content of the signal before passing it to the P2075 amplifier and cockpit speakers. This mixer also accepted a third separate audio channel from the EPS on which the avionics tones and alarm effects had been isolated. It's #1 *effects send* was used as a third output to patch the signals directly into the pilot's intercom set. The third mixer was configured exactly the same, but was used to supply the LAMARS with effects from it's own EPS tone generator. For the MCS cockpits no mixer was required since a full pre-mix of environmental effects and avionics cues come directly from the EPS.

There are actually three independent sound systems in the WL Engineering Flight Simulation Facility: one for MS-1, one for LAMARS and a third for the MCS cockpits. The MS-1 and LAMARS simulators each use a dedicated EPS, providing a full 20-voice sound effects capability. The four MCS cockpits share an EPS, dividing it's 20 voices among four virtual samplers.

The 20 EPS voices are allocated dynamically, meaning that a single MCS can use more than five voices simultaneously (in fact, up to 20). However, the total number of voices playing at one time for all MCS's can never total more than 20.



WL Engineering Flight Simulation Facility Sound Effects Systems

Looking Ahead

One of the fastest growing segments in the world of simulation is virtual reality. Without getting into a discussion of what virtual reality is, it is worth mentioning how virtual reality developments in 3-D sound might apply to air-combat simulation. 3-D sound, for lack of a better definition, is an attempt to present sound to a listener in such a fashion that the sound can be localized in a three dimensional space.

The first question most people ask when the subject of 3-D audio and air-combat simulation comes up is why is 3-D audio necessary? And the answer is simply that 3-D audio is not necessary in most cases to model the sonic environment of the cockpit as we know it today. However, it does not mean that 3-D sound cannot be used to generate artificial cues to aid the pilot in air-combat.

The Armstrong Aerospace Medical Research Laboratory is currently investigating the use of 3-D sound as an *auditory display*. As part of their virtual reality cockpit, the pilot is presented up to four 3-D audio cues, each localized relative to the pilot's ear points. For example, audio cues can be assigned to represent other aircraft in the simulation. Although the pilot may or may not be able to see other aircraft, he can constantly *hear* their location and by judging the volume of the cue, determine the target's relative distance. Knowing the location of other aircraft without having search out the window or look down and interpret a HDD could significantly reduce pilot workload and increase situational awareness. A of 3-D sound system like this could enhance low cost cockpit stations with restricted visuals. At the WL Engineering Flight Simulation Facility, this could help to reduce the disparity between the capabilities of the MCS cockpits and the higher-performance MS-1 and LAMARS simulators.

While 3-D sound is progressing, it has not been perfected. Due in part to anatomical differences in the head and ears, many people have difficulty localizing sound produced by these systems[5]. Why some people can localize sound from these machines and others cannot is beyond the scope of this paper. However research continues to improve the effectiveness of 3-D sound systems.

Conclusion

Applying digital music instrument technology to simulator sound effects generation provides numerous pay-offs for the WL Engineering Flight Simulation Facility. Foremost are the direct pay-offs for the ICAAS program. A host of additional advantages for the facility and the Air Force at large are only straightforward implementations away.

For ICAAS, aural cues are an important part of the air-combat environment. Along with good visual cues, they significantly enhance the overall combat-simulation environment and add an extra degree of realism. Pilots have reported a tangible increase in their belief in battle scenarios when sound effects from this system were used. The increased ability to suspend disbelief can only result in more realistic pilot response to battle stimuli which in turn ultimately translates into more valid simulation tests.

It can safely be said that the sound system met all of the ICAAS program requirements and in many cases far exceeded them. Initially, it was expected that roughly $50K per cockpit would be required to outfit the simulators with a sound effects capability that would meet ICAAS requirements. In fact, had a more traditional approach been taken this would indeed have been the case. Using sampling synthesizer technology and a little creative engineering brought that cost down to about $15K per cockpit ($45K total) while at the same time providing the facility with a far more flexible simulation tool.

An additional benefit was realized when the sound system was completed ahead of schedule. Other simulation programs being conducted at the facility were able to use the system before it would be needed for ICAAS. These programs effectively provided a hot-test bed for the new capability which proved to be invaluable. The development of simulation host control software for the sound-system benefited in particular as these programs provided a steadily increasing demand on the sound-system's capability with a commensurate increase in the complexity of the control software.

Although produced for ICAAS and driven by it's requirements, the resulting sound system is becoming a well used long-term facility-wide resource. By it's very nature, this adaptable system compliments the WL Engineering Flight Simulation Facility. With each new simulation program demanding its own unique set of requirements, the ability to customize simulation tools quickly and without great expense becomes an important issue. The system's demonstrated flexibility and ability to rapidly prototype new sound effects sets help keep operational costs down. And since each major component of the system is strictly off-the-shelf commercial gear, maintenance costs will stay low. For these reasons, many varied aircraft assessment simulation programs will continue to use the sound system well into the future.

As a final pay-off, this system can be gracefully expanded to meet even the most demanding facility requirements simply by adding additional EPS sound generators and audio reinforcement. Today with the non-recurring engineering completed, the estimated cost to expand this system is under $7K per cockpit. Existing control software will accommodate up to sixteen cockpits without any modification, and will support a practically unlimited number of cockpits with only minor modification.

Searching for low cost/high return alternatives to simulator subsystem acquisition is a noble cause. Implementing such an alternative requires engineers and managers to take risks and be creative. Today, in the face of shrinking budgets, one's duty should require no less!

References

[1] MIDI 1.0 Detailed Specification, The International MIDI Association, Los Angeles, CA, 1989.

[2] EPS-m Musician's Manual, ENSONIQ Corporation, Malvern, PA, 1988.

[3] Mazanowski J., "A MIDI-Based Aural Simulation System", Proceedings of the Inter-Service/Industry Training Systems Conference, Orlando, FL, 1990

[4] Lee, Burnette. Personal Interview, 21 January 92.

[5] Haas, Michael. Personal Interview, 29 May 92.

AIAA-92-4167-CP

An Electronic Visual Display Attitude Sensor (EVDAS) for Analysis of Flight Simulator Delays

Gary Jeff Slutz
Systems Engineer
Electronic Associates Inc.
Bldg 145; Area B
WPAFB, OH 45433

Ronald B. Ewart
Principal Scientist
WL/FIGD
WPAFB, OH 45433

Abstract

This paper discusses a new approach for the measurement of visual system time delays for flight simulators. The approach involves the use of an electronic circuit which monitors the video in raster format going into the pilot's display subsystem. The circuit measures the horizon's pitch and roll angles and outputs the information so that it can be used for time delay analysis. Time delay measurement techniques are reviewed and several practical applications are discussed. Experimental data for an existing simulation and techniques for minimizing delays based upon the results are discussed.

Problem

The synchronization of cues presented to a simulation pilot has historically been a problem[2, 3, 5 & 7]. In order to be effective, the pilot's stick input must cause the properly time phased responses by the aircraft's simulated instrumentation, motion base, and visual displays. Use of computer generated visual systems adds at least 50 ms. and often considerably more to the total transport delay of any simulation system. This visual transport delay must be minimized and properly interfaced to the total simulation. The transport delay has previously been measured using a variety of methods including placing a photo cell on the pilot's visual display screen or monitoring the video at one point in the scene, and detecting a step change in attitude[1, 4 & 6]. The technique discussed in this paper improves upon theses existing techniques by providing a practical means of measuring pitch and roll angles of a horizon or HUD pitch ladder directly from the video.

An Electronic Visual Display Attitude Sensor (EVDAS) was designed and tested at the Flight Control Division of Wright Laboratories at Wright Patterson Air Force Base (WPAFB), OH. The EVDAS generates two electronic video sensor stripes which are positioned to detect horizon or picture element movement near the left and right side of the picture. By detecting movements of either the horizon or a picture element such as the pitch ladder on the Heads Up Display (HUD), a digital number and resulting analog voltage proportional to the pitch or roll angle is generated. This voltage can be used to compare with a pilot stick input signal or test signal output to determine total simulation delays. It can also be compared with the simulation's output to the visual system to determine time delays due to the visual system interface and the Computer Image Generator (CIG).

Time delays which are introduced by computer generated image systems have always been a concern for high fidelity simulations. While many of the newer visual systems have inherently acceptable time delays, they must be properly synchronized with the simulation and input/output routines in order to take full advantage of their reduced transport delay. The EVDAS provides a means of doing this by measuring the final resulting display response as seen by the pilot. It is not uncommon to uncover an unknown timing latency by measuring the end-to-end transport delay of the simulation system using the EVDAS. The EVDAS can also be used to optimize the response of synchronous CIG systems using a technique discussed later in this paper.

Electronic Visual Display Attitude Sensor (EVDAS)

The EVDAS uses a novel electronic circuit which monitors the video in raster format going into a pilot's display subsystem. It measures and outputs horizon pitch or roll angles of the horizon for real time analysis. The unit can also be used to measure pitch and roll angles for graphical images. It has been successfully used to measure the pitch and roll angles for displayed HUD imagery.

The EVDAS is typically connected to the video display and data recording equipment as shown in figure 1. For purposes of simplification, this discussion will only refer to sensing of a blue sky and horizon although the unit can be used to sense a HUD's pitch ladder position and angles as well. The EVDAS electronically generates two moveable vertical sensing stripes or windows in video. One stripe is placed near the left side of the displayed image and one is placed near the right. These electronic stripes are simply gating pulses in the EVDAS which tell the unit when to sense for the presence or absence of the blue sky. Figure 2 illustrates the primary EVDAS controls and the resulting test monitor display which simplifies setup of test experiments.

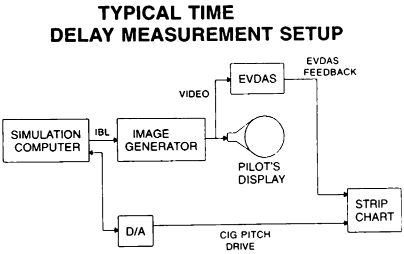


Figure 1. Typical Time Delay Measurement Setup

The EVDAS uses threshold detection of the clamped video to determine whether a color such as blue sky is present or absent. See figure 3 for the EVDAS functional block diagram. A video threshold adjustment is provided for setup. The unit outputs a buffered video level which represents the detected video. This signal is fed into the blue channel of a test monitor. The test monitor provides a quick way of setting up the threshold; the video threshold is adjusted until the detected blue sky is displayed with a crisp separation from the ground below.

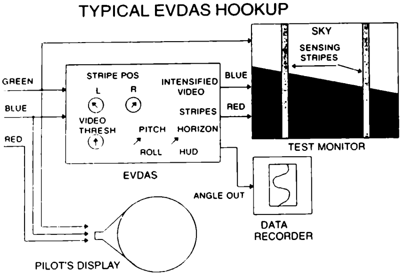


Figure 2. Typical EVDAS Hookup

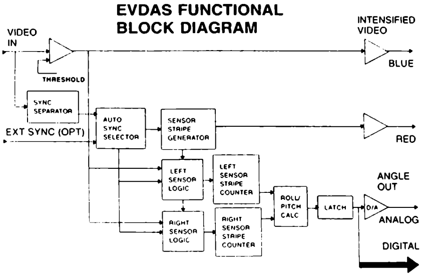


Figure 3. EVDAS Functional Block Diagram

Video sync is required for the internal reference timing of the EVDAS. It is obtained from either an external sync source or by stripping sync from the composite green video input. An automatic sync selector defaults to external sync if provided or uses stripped sync if external sync is absent. The horizontal and vertical components of the sync are separated and used by the EVDAS' internal logic.

For use with a simple sky and horizon, the EVDAS counts each time a video line, depicting sky, crosses the left and right sensor stripes. The left and right sensor stripe counters keep track of the crossings during each video field time. The counters' outputs are transferred to the roll or pitch calculator at the end of each field. The sensor stripe counters are reset to zero during the vertical interval and the sensing counter cycle begins again on the first active video line of the next field.

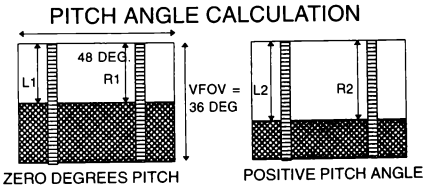
The roll and pitch calculator takes the outputs from the sensor stripe counters and determines a number related to the pitch or roll angles of the displayed horizon. In the pitch mode the average of the left and right sensor stripe counters is output; in the roll mode the difference between the two counters is output.

As an example of the pitch and roll calculator's operation, assume that a standard 1023 line video image depicting terrain, horizon and sky is to be analyzed. Assume that this is a typical simulator video display with a horizontal field of view of 48 degrees and a vertical field of view of 36 degrees. There are 959 total active lines per frame, or 479.5 active lines in each 16.66 ms. field. See figure 4. Testing is to be done in the longitudinal aircraft axis; the pitch axis has been selected on the EVDAS. The left illustration in figure 4 shows the horizon at 0 degrees. Both the left and right sensor stripe counters are reset to 0 during the vertical interval. They both start at 0 at the top of each field. As the picture is displayed and is scanned out from top to bottom, the EVDAS counts the number of lines where the sky is present underneath each sensor stripe. Assume that the horizon at zero degrees pitch coincides with the middle of the video picture. Under these conditions the left sensor stripe counter would register 0.5 * 479.5 or 240. For a level horizon, the right sensor stripe counter would also register 240. In the pitch mode, the EVDAS adds the two counters and divides by 2. The resultant count is 240 representing 0 degrees pitch. If the pilot pitches up as illustrated in the right hand illustration in figure 4, more sensor stripe lines are covered by the sky. Again the left and right sensor line counters are added together and divided by 2 to get a number related to the pitch angle.

Roll data is determined in a similar manner with the exception that the difference of the sensor stripe counters is output by EVDAS. See figure 5.

The EVDAS uses the active field time to count the number of raster lines appropriate for either the pitch or roll measurement. At the beginning of the vertical interval, the sensor stripe counters transfer the result into a data latch. This data is immediately available to the external measurement device. The data is simultaneously output as both a digital number and an analog voltage. Either of these outputs can be used for analysis depending upon the test setup. The digital numbers represent the number of TV lines crossed by the sensing stripes. Analog voltages are proportional to the digital data. The EVDAS output is accurate to a resolution of one TV line.

The timing of the digital data valid strobe and the transition of the analog output voltage results are consistent with the accepted definitions for transport delay. Most definitions for visual system transport delays require that the entire video field be completely displayed.



LET:

N = NUMBER OF ACTIVE TV LINES / FIELD

L = NUMBER OF LINES DETECTED BY LEFT SENSING STRIPE

R = NUMBER OF LINES DETECTED BY RIGHT SENSING STRIPE

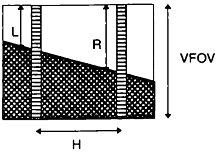
VFOV = VERTICAL FIELD OF VIEW IN DEG

THEN:

$$\Theta = \frac{(R2 + L2) - (R1 + L1)}{2} * \frac{VFOV}{N}$$

Figure 4. Pitch Angle Calculation

ROLL ANGLE CALCULATION



LET:

VFOV = VERTICAL FIELD OF VIEW IN DEG

H = HORIZONTAL SPACING BETWEEN SENSOR STRIPES IN DEGREES

N = NUMBER OF ACTIVE TV LINES / FIELD

L = NUMBER OF LINES DETECTED BY LEFT SENSING STRIPE

R = NUMBER OF LINES DETECTED BY RIGHT SENSING STRIPE

THEN:

$$\phi = \mathrm{ARCTAN}\left[\frac{(R - L) * (VFOV / N)}{H}\right]$$

Figure 5. Roll Angle Calculation

The derivation of the pitch and roll angle equations are shown in figures 4 and 5 respectively. In practical experiments using relatively small pitch and roll angles, the data coming out of the EVDAS can be scaled and used directly to represent angles with good results. The experimental results shown later in this paper are all taken directly from the output of the EVDAS.

HUD Capability

A simulator's background terrain image must be properly correlated with the HUD's pitch ladder. The background and HUD images are often simulated with different image generators. For example the background terrain may be generated by a real time image generator and the HUD may be simulated with a graphics workstation. Improper correlation of the two images results in a poor simulation due to the conflicting attitude information presented to the pilot during high rate maneuvers. It is desirable to use the EVDAS to detect movements of graphical images such as the 0 degree line on a HUD's pitch ladder.

The EVDAS can be used to detect either horizon or graphical movement; a toggle switch on the device makes the selection. In the typical horizon mode, the EVDAS simply counts the number of lines where the sensor stripes lie on top of the sky. In the HUD or graphical mode, both the left and right counters are reset to zero and enabled at the start of each field. All sensor stripe line crossings are counted until the sensor stripe crosses the graphical line of interest. In typical experiments the color of the zero degree HUD pitch ladder is changed to blue; the blue color makes the line easy for the EVDAS to separate out from the rest of the pitch ladder. The sensor stripes are also moved away from the edges of the scene so that they cross the pitch ladder symbology.

One deficiency of the EVDAS is that it cannot handle stroke or calligraphic generated HUD symbology. This is not a problem in the Flight Dynamics' flight simulator facility because Silicon Graphics raster image generators are used to create the HUD symbology.

Potential Future Embellishments

Using the current straight vertical sensor stripes, the roll angle measurements are restricted to those where the horizon crosses within the field of view covered by the stripe. For the 48 by 36 degree example discussed above, roll angle measurements are restricted to about 36.8 degrees for widely separated sensor stripes which provide the greatest angular resolution. This is adequate for most measurements; however, other shapes of sensor stripes could be used to extend the useful range of the EVDAS in the roll mode. For example, a circular or elliptical sensor stripe could be used to generate the sine and cosines of roll angles. This technique would permit full 360 degree roll angles to be measured by the EVDAS.

Improvements in detected color sensitivity could be made by using color combination keying. In practice, the EVDAS has been able to easily and accurately detect the horizon using single color only threshold detection.

Experiments

The Electronic Visual Display Attitude Sensor's (EVDAS) capability opens the door for new and improved simulator visual system testing techniques. Experiments were conducted on image generators at the Flight Dynamics Directorate of Wright Labs. General Electric's Compu-scene C-IVa, and various models of the Silicon Graphics image generators including the 4D-85GT and W-4D-340VGX were used for these tests. The new testing techniques which were tried using the device are described below.

Transport Delay Minimization

The EVDAS was used to measure the C-IVa's transport delay of both a standard and a reduced transport delay databases. These measurements were taken with the simulation running both synchronously and asynchronously. Data gathered during the synchronous testing was then used to develop a method for minimizing the transport delay by adjusting the relative timing of the C-IVa and the simulation computer.

When running synchronously with the C-IVa image generator, it is possible to incur transport delays that range from the nominal value of 77 ms. to approximately 94 ms. The variation is due to the relationship between when the data arrives at the C-IVa and when the next frame-one cycle of the C-IVa begins. The C-IVa is functionally divided into three sections called "frames". Frame one is the first section of the pipeline and has as one of its functions the handling of the host interface. As shown in figure 1, the interface between the simulation computer and the C-IVa is via an HSD IBL (High Speed Data Interbus Link). Simulation data can be deposited into the C-IVa memory any time during a frame-one cycle because the HSD is a DMA type device. In the synchronous mode of operation, the simulation frame start interrupt is generated by the C-IVa at a 60 hz rate. The C-IVa uses a term called HPS or Host Programmable Sync Interrupt to adjust this time. It varies the host interrupt time relative to frame-one start; adjustments can be made to fractions of milliseconds. Thus the start of the simulation frame can be adjusted so that the data arrives at the C-IVa just prior to the beginning of the next frame-one cycle which minimizes the need for interpolation via rate data. Figure 6 shows the above mentioned timing relationship.

At Wright Labs a method has been developed to correctly adjust this C-IVa parameter under actual simulation conditions. The correct setting of the C-IVa parameter is a function of the time between the beginning of the simulation frame and the time in the frame when data is transferred to the C-IVa. Thus the correct setting can vary from simulation to simulation or from run to run. A switch in the software can be set which causes the C-IVa data transfer routine

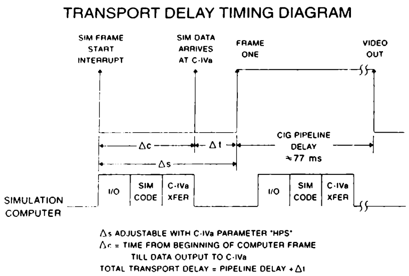

TRANSPORT DELAY TIMING DIAGRAM
SIM FRAME START INTERRUPT
SIM DATA ARRIVES AT C-IVa
FRAME ONE
VIDEO OUT
Δc
Δt
Δs
CIG PIPELINE DELAY ≈77 ms
SIMULATION COMPUTER
I/O
SIM CODE
C-IVa XFER
I/O
SIM CODE
C-IVa XFER
Δs ADJUSTABLE WITH C-IVa PARAMETER "HPS"
Δc = TIME FROM BEGINNING OF COMPUTER FRAME TILL DATA OUTPUT TO C-IVa
TOTAL TRANSPORT DELAY = PIPELINE DELAY + Δt

Figure 6. Transport Delay Timing Diagram

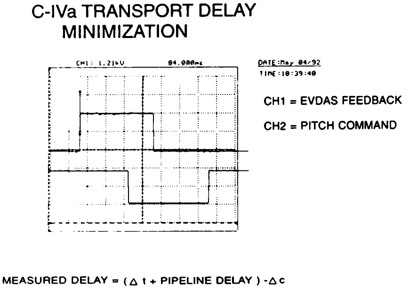


Figure 7. C-IVa Transport Delay Minimization

to step the value of pitch in an alternating manner. This same pitch value is output via a D/A at the beginning of the next simulation frame. The D/A output and the output of the EVDAS are observed on an oscilloscope while the C-IVa parameter is varied. Figure 7 shows a typical oscilloscope presentation during this procedure. The goal is to adjust the C-IVa parameter so that the observed time difference between the D/A output and the EVDAS output is minimized. Note that the observed time difference on the oscilloscope does not represent the actual transport delay due to the one frame delay in the D/A output. However, minimizing the observed value will guarantee minimization of the actual transport delay. This procedure has been verified on simulations running different length schedules (real time code). Figure 8 is a plot of delay versus "HPS" for one such case. The execution time of the code as measured from the beginning of the simulation frame to the end of the C-IVa transfer routine is also annotated on this figure. It should be noted that as the optimal value of HPS is approached, the measured delay may jump between a "minimum" and a "maximum" value. This is labeled on figure 8 as the "min/max transition zone". To insure the minimum constant transport delay, HPS should be set so that the measured delay is just outside this transition zone. The procedure discussed above is applicable to any simulation running synchronously with a constant transport delay image generation system and demonstrates a very practical use of the EVDAS.

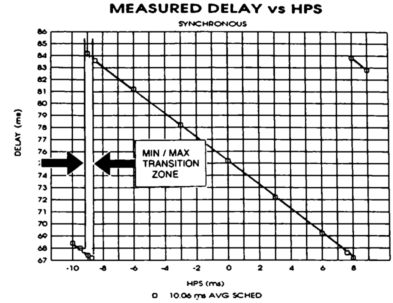


Figure 8. Measured Delay vs HPS

Transport Delays - - Synchronous Real Time vs Asynchronous Graphic Workstation Image Generators

It is interesting to compare two image generators with and without synchronous operation and load management features. For our comparison we used the General Electric Compu-scene C-IVa and the Silicon Graphics 4D-85GT image generators. The Compu-scene represents an image generator with both synchronous operation and a load management system. It predictably completes the creation and display of all pictures at a uniform rate. The Silicon Graphics 4D-85GT has neither synchronous operation nor a load management system. It therefore has a large variation of transport delay times depending upon the complexity of the scene and arrival time of data from the host computer.

To illustrate the characteristics of both types of image generators, an image was created which contained an austere scene with horizon and sky and an aircraft image which periodically entered into the scene. A sinusoidal drive was output to each image generator which drove the pitch position of the horizon. The EVDAS was used to measure the pitch position as a function of time. The C-IVa consistently output the scene at a constant update rate and with a constant transport delay. The transport delay was always 77 ms. and the update period was always 16.6 ms. See figure 9 for a

CIG TRANSPORT DELAYS

SYNCHRONOUS REAL TIME vs ASYNCHRONOUS GRAPHICS BASED MACHINES

IMAGE GENERATOR	HORIZON + AIRCRAFT		AUSTERE HORIZON	
	TRANSPORT DELAY (ms)	UPDATE PERIOD (ms)	TRANSPORT DELAY (ms)	UPDATE PERIOD (ms)
C-IVa	77	16.6	77	16.6
SILICON GRAPHICS 4D - 85GT	135 to 170 148 AVG	83.3 to 100 100 MEDIAN	90 to 130 109 AVG	50 to 83.3 50 MEDIAN

Figure 9. CIG Transport Delays;
Synchronous Real Time vs Asynchronous Graphic Based Machines

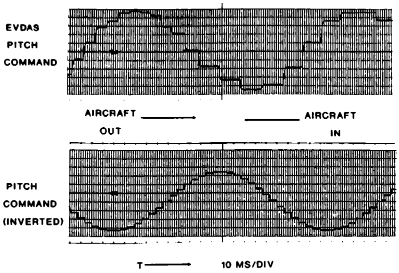


Figure 10. Silicon Graphics Time Delays with Varying Scene Complexity

tabulation of the test results. The Silicon Graphics image generator showed variable delay effects due to both asynchronous operation and image complexity. Figure 9 also shows the results of the same test for the Silicon Graphics 4D-85GT machine. With only an austere horizon being displayed, the transport delay varied from 90 to 130 ms and the update period varied from 50 to 83.3 ms. When the aircraft is displayed in the picture also, the image generator must draw both the austere horizon and the aircraft image. All polygons must be drawn which slows down the image generation and results in a transport delay which varies from 135 to 170 ms. with an update period which varies from 83.3 to 100 ms. See figure 10.

The illustration above becomes especially significant for simulations which use two different machines with different capabilities to draw background scenery and HUD imagery. It is easy to see potential problems resulting from non-correlated background and HUD transport delays under dynamic conditions.

C-IVa Extrapolation

Testing was conducted to determine how and when extrapolation took place on the C-IVa image generator. It was unclear at the time whether the C-IVa performed extrapolation only if new data was not presented every frame-one cycle (this would happen under asynchronous operations) or whether it would also extrapolate from the time of data arrival to the next frame-one cycle under synchronous operating conditions. To test the latter case a simulation was run in synchronous mode utilizing various rate data and position data combinations. First the HPS parameter was adjusted on the C-IVa so that the data would arrive approximately 6 ms. before the beginning of the next frame-one cycle. The HPS parameter controls the timing relationship between the simulation start frame pulse and the beginning of the frame-one cycle. Next, pitch was stepped between 0 and 10 degrees and sent to the C-IVa with no rate information. This was done to establish the EVDAS levels for 0 and 10 degrees, and to determine the proper degrees per volt scaling. Next, the same pitch step was sent but this time with a -500 degrees/sec pitch rate. If extrapolation was taking place the expected angles as indicated by EVDAS would be -3 degrees and 7 degrees; 6 ms. x -500 deg/sec = -3 deg. As shown in figure 11 the indicated values were quite close to the expected values assuming extrapolation. The test was repeated with a pitch rate of -1000 deg/sec and similar results were obtained indicating that the C-IVa does indeed extrapolate synchronous position data to the next frame-one cycle in the presence of rate data.

The capability of the EVDAS to provide an output representative of the angle being displayed was key to this testing.

C-IVa EXTRAPOLATION TEST RESULTS

INPUT STEP (degrees)	RATE (deg/sec)	EVDAS (volts)	EVDAS (degrees)
0 to 10	0	4.8 to 7.3	0 to 10
0 to 10	-500	4.0 to 6.5	-3.2 to 6.8
0 to 10	-1000	3.3 to 5.7	-6.0 to -3.8

Figure 11. C-IVa Extrapolation Test Results

Frequency Domain Measurement

Experiments were conducted to demonstrate the capability of the EVDAS in terms of frequency domain measurements. A given continuous domain transfer function, $\frac{625}{s^2 + 35s + 625}$, was discretized by using the Tustin transform with prewarping at 25 rad/sec resulting in a z transform of $\frac{0.033\,(1 + 2Z^{-1} + Z^{-2})}{1 - 1.42Z^{-1} + 0.544Z^{-2}}$. Figure 12 shows the setup for this test. A test signal (white noise or swept sine wave) was applied to the simulation via an A/D and simultaneously to one channel of an HP5420A frequency analyzer. This test signal served as the input to the transfer function while the output of the transfer function was used to drive the C-IVa in pitch. The EVDAS was connected to the video from the C-IVa while its output was connected to the other channel of the frequency analyzer. The analyzer was set up for a transfer function measurement with a bandwidth of 12.5 hz. Figure 13 shows a comparison between the bode plot obtained utilizing white noise and the theoretical bode plot. Figure 14 shows a comparison between the bode plot obtained with a frequency sweep and the theoretical bode plot. Additional phase lag due to transport delay of the C-IVa and the zero order hold have been factored into the theoretical/expected phase plots. Both figures show close agreement between the measured and theoretical bode plots in both phase and magnitude.

The EVDAS updates its output at the end of a field and holds that value until the end of the next field. Thus the EVDAS is similar to a D/A running at a 60 hz rate and is subject to errors due to quantisization and errors due to the zero order hold. However, the capability of the EVDAS to provide sufficient information to make frequency domain measurements is clearly shown in the figures. This capability becomes especially significant when evaluating the effectiveness of compensation schemes in the frequency domain.

FREQUENCY DOMAIN TEST SETUP

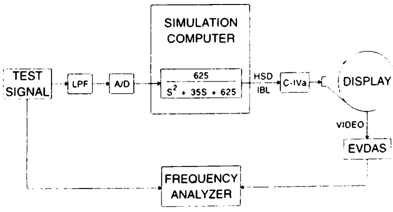


Figure 12. Frequency Domain Test Setup

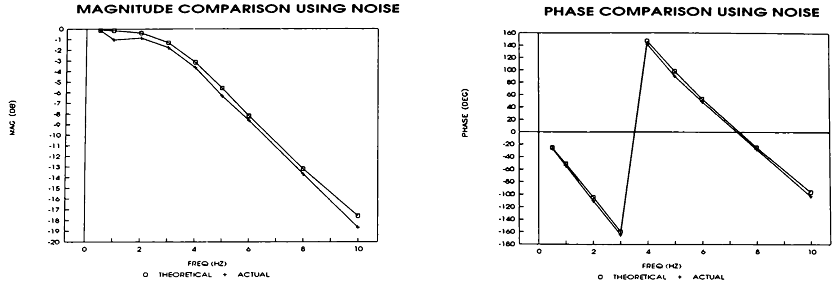


Figure 13. Bode Plots with Noise Input

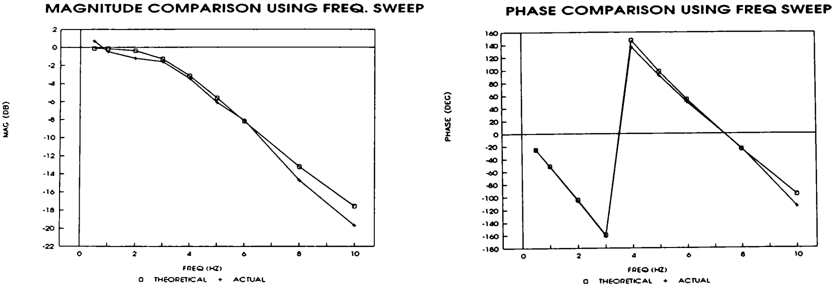


Figure 14. Bode Plots with Swept Sine Wave Input

Correlation / End-to-End

The purpose of the correlation/end-to-end experiment was to measure the end-to-end delay from stick input to visual output in the time domain. The general method employed was to sample the stick input and the output data from the EVDAS with a separate computer system and store this data to a file for post processing. A correlation routine was then applied to this data to obtain the transport delay. Figure 15 shows the setup for this experiment. Computer system "C" was the host machine for the simulation and ran at a 60 hz rate synchronous with the C-IVa. Computer system "A" served as the sampling system and ran at a 500 hz rate. The stick input was applied to a "dynamics" routine which had three software selectable modes of operation. In mode one the input was considered to be a pitch position command and thus only scaling was applied prior to output to the C-IVa. In modes two and three the input was considered to be a pitch rate command and thus was integrated and scaled prior to output to the C-IVa. The difference between modes two and three are the methods of integration. Mode two utilizes the trapezoidal method while mode three utilizes the second order Adams-Bashforth method. The motivation behind the two different integration methods was to show the effects of the integration method on the measured delay. Under steady state conditions it was expected that the measured delay for the Adams-Bashforth method would be less than that of the trapezoidal method due to the predictive nature of the Adams-Bashforth method. Testing was conducted using square waves and sin waves of various frequencies from a signal generator and actual stick input. In the cases where the input represented a position command a cross correlation was performed directly on the sampled input data and the sampled EVDAS output. In the cases where the input represented a rate command, the sampled input data was first integrated and then a cross correlation was performed on this integrated signal and the EVDAS output data. Since the input data was sampled at a 500 hz rate and the input frequencies were less than 2 hz, this off-line integration of the input data produces results very close to that of an ideal integrator. Figure 16 shows the results of this testing. The last column shows the transport delay as determined from the correlation data. Under the given test conditions the transport delay is represented by the time of the first maximum in the correlation plot. Figure 17 shows the raw data and the corresponding correlation plot for the case of a square wave input to a trapezoidal integrator. Figure 18 shows the raw data and the corresponding correlation plot for the case of a 2 hz sin wave input to an Adams-Bashforth integrator. The two different correlation plots represent two different

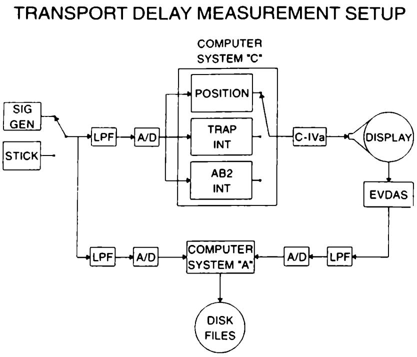


Figure 15. End-to-end Transport Delay Measurement Setup

CORRELATION TEST RESULTS

INPUT	MODE	DELAY (ms.)
Square wave	Position	90
Square wave	Trap integrator	88
Square wave	ab2 integrator	74
2 hz. sin	Position	90
2 hz. sin	Trap integrator	90
2 hz. sin	ab2 integrator	74
Stick	Position	90
Stick	Trap integrator	90
Stick	ab2 integrator	76

Figure 16. Correlation Test Results

algorithms used to calculate the cross correlation. The predicted measured value for transport delay was approximately 88 ms. This value is a combination of the average value of the sampling uncertainty (8.3 ms.), the advertised transport delay of the C-IVa (77 ms.), and the delay due to the low pass filters on the front end of the A/D converters (approx. 2 ms. to 3 ms.). The measured value was very near the predicted value for the position command case and the trapezoidal integrator case. The delay when using the Adams-Bashforth method is less by almost one simulation frame due to its predictive nature.

The goal of the above testing was to evaluate the capability of the EVDAS in terms of time domain testing and to lay the groundwork for future efforts in this area.

Future Plans

Future plans for the EVDAS includes extending the correlation test which determines total day under dynamic conditions. Up to this point the EVDAS has been used to correlate test signals such as pulses, square wave, and sine wave inputs with the visual display output. A limited amount of random stick input has been used as the input source. These test results are very encouraging and indicate that transport delays could be measured directly using pilot stick inputs during simulated missions. Planned testing will attempt to correlate pilot stick inputs with the EVDAS output to determine actual total delay.

We have also proposed to use a modified version of the EVDAS combined with a simple PC based network monitoring device using GPS to measure long haul networked simulators' time delays. These measurements could provide valuable insight into the total time delays between stick input in one simulator until the simulator's aircraft image is perceived to move in the second networked simulator.

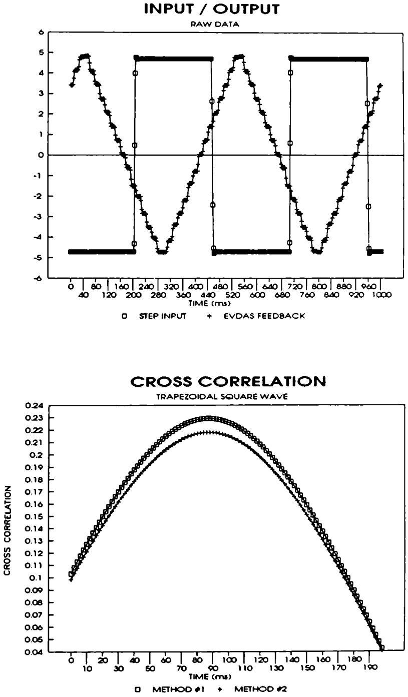


Figure 17. Square Wave Input Test; Trapezoidal Integration

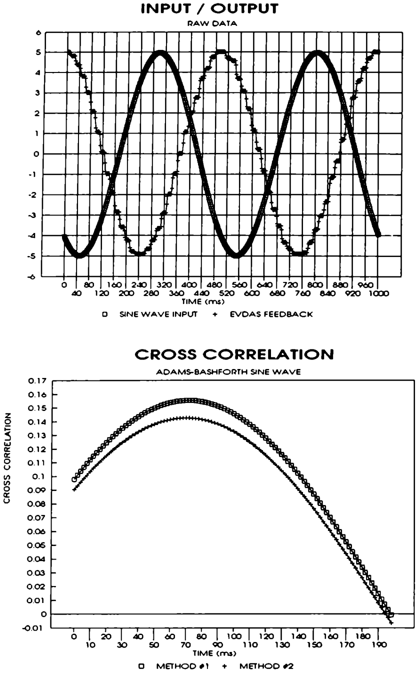


Figure 18. Sine Wave Input Test; Adams-Bashforth Integration

Conclusion

Direct video analysis can be successfully used for a variety of flight simulator timing measurements in both the time and frequency domains. The EVDAS described by the paper allows direct measurement of the video signal going to the pilot's display; the results measured by the device correspond exactly in time with the picture which the pilot views. It allows visual system time delays to be measured rather than predicted. Techniques developed using the EVDAS permit transport delays to be minimized for computer image generators which operate in the synchronous mode. The EVDAS is also very useful in locating unexpected delays due to implementation. Predicted delays often miss subtle time delays especially in multiprocessor simulation environments. The EVDAS can verify predicted delays or pinpoint causes of unknown delays. The EVDAS extends the practical range of time delay analysis and measurements which can be done on flight simulators.

References

1. Smith, R.M., "A Method for Determining Transport Delays in the Flight Simulation Environment", AIAA Paper No. 91-2964-CP, August 1991.

2. McMillan, G.R., "Cue Integration and Synchronization", January 1990.

3. Howe, R.M., "Some Methods for Reducing Time Delays in Flight Simulators", AIAA, 1990.

4. Johnson, W.V. and Middendorf, M.S., "Simulator Transport Delay Measurement Using Steady-State Techniques", AIAA Paper No. 88-4619, September 1988.

5. Gum, D.R. and Martin, E.A., "The Flight Simulator Time Delay Problem", AIAA Paper No. 87-2369, August 1987.

6. Ewart, R.B., "Time Delay Measurements for Flight Simulators", SCSC Paper, July 1977.

7. Gum, D.R. and Albery, W.A., "Time-Delay Problems Encountered in Integrating the Advanced Simulator for Undergraduate Pilot Training", J. Aircraft, 1977.

AIAA-92-4168-CP

SPECIAL ROTATION VECTORS QUATERNIONS IN THREE COMPONENTS

Amnon Katz[*]
The University of Alabama
Tuscaloosa, AL 35487-0280
akatz@ua1vm.ua.edu

ABSTRACT

Special Rotation Vectors (SRV) are a means for specifying orientation in three dimensions that is a cross between a generalization of Wigner's rotation vectors and Euler/Hamilton quaternions. SRV constitute a continuous one to one mapping of orientation over a three dimensional manifold. Orientations are specified by three parameters while retaining virtually all the computational advantages of the quaternion method.

I OVERVIEW

The geometry of orientation in three dimensions and the associated kinematics are a subject of considerable complexity [1]. Unlike translations, rotations around mutually perpendicular axes do not commute; rather they affect each other. This manifests itself in flight maneuvers such as the *Immelmann* and the *Split-S* that combine pitch and roll to reverse heading. During the first half of a *chandelle*, bank increases all the way to 90° by pitching alone, whereas a *lazy eight* involves a phase where yawing alone affects both pitch and bank.

The foundations of orientation analysis were laid by Leonhard Euler circa 1705 [2,3,4]. The Euler angles used in aeronautics are easier to grasp than the set introduced by Euler and still prevalent in texts on analytic mechanics. Heading, pitch, and bank have immediate intuitive meaning for the pilot and the engineer. Nevertheless, in the context of computer applications that determine orientation by step-wise integration, the formalism of Euler angles leaves something to be desired. Equations (1) and (2) exhibit the flaws. (1) is the construction of a rotation matrix (for an active rotation) from Euler angles. Equation (2) expresses the time derivatives of Euler angles in terms of (body axes) components of angular velocity.

$$R = \tag{1}$$

$$\begin{vmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\varphi - \sin\psi\cos\varphi & \cos\psi\sin\theta\cos\varphi + \sin\psi\sin\varphi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\varphi + \cos\psi\cos\varphi & \sin\psi\sin\theta\cos\varphi - \cos\psi\sin\varphi \\ -\sin\theta & \cos\theta\sin\varphi & \cos\theta\cos\varphi \end{vmatrix}$$

$$\left|\begin{aligned} \dot\psi &= (\omega_2\sin\varphi + \omega_3\cos\varphi)/\cos\theta, \\ \dot\theta &= \omega_2\cos\varphi - \omega_3\sin\varphi, \\ \dot\varphi &= \omega_1 + (\omega_2\sin\varphi + \omega_3\cos\varphi)\tan\theta, \end{aligned}\right. \tag{2}$$

(Components of ω in body axes).

When $\theta=\pm\frac{1}{2}\pi$, $\dot\psi$ and $\dot\varphi$, as given by equation (2) become infinite. This is related to the failure of Euler angles to provide a one to one continuous mapping of aorientation. For $\theta=\pm\frac{1}{2}\pi$, many combinations of heading and bank represent the same orientation, and large excursions in these variables are possible with no change in orientation. The need to compute six different trigonometric functions at each integration step, evidenced in both (1) and (2), is also undesirable.

An alternate formalism for computing the history of orientation, free from these objections, was also provided by Euler [4]. This method, which he called "the four parameter method" uses four quantities Q_0, Q_1, Q_2, Q_3 to represent orientation. The counterparts of equations (1),(2) are (3),(4).

$$R = \tag{3}$$

$$\begin{vmatrix} Q_0^2 + Q_1^2 - Q_2^2 - Q_3^2 & 2Q_1Q_2 - 2Q_3Q_0 & 2Q_1Q_3 + 2Q_2Q_0 \\ 2Q_2Q_1 + 2Q_3Q_0 & Q_0^2 - Q_1^2 + Q_2^2 - Q_3^2 & 2Q_2Q_3 - 2Q_1Q_0 \\ 2Q_3Q_1 - 2Q_2Q_0 & 2Q_3Q_2 + 2Q_1Q_0 & Q_0^2 - Q_1^2 - Q_2^2 + Q_3^2 \end{vmatrix}$$

[*] Professor of Aerospace Engineering, Member AIAA.

$$\left|\begin{aligned} \dot{Q}_0 &= -\tfrac{1}{2}Q_1\omega_1 - \tfrac{1}{2}Q_2\omega_2 - \tfrac{1}{2}Q_3\omega_3, \\ \dot{Q}_1 &= -\tfrac{1}{2}Q_0\omega_1 - \tfrac{1}{2}Q_3\omega_2 + \tfrac{1}{2}Q_2\omega_3, \\ \dot{Q}_2 &= +\tfrac{1}{2}Q_3\omega_1 - \tfrac{1}{2}Q_0\omega_2 - \tfrac{1}{2}Q_1\omega_3, \\ \dot{Q}_4 &= -\tfrac{1}{2}Q_2\omega_1 + \tfrac{1}{2}Q_1\omega_2 - \tfrac{1}{2}Q_0\omega_3, \end{aligned}\right. \tag{4}$$

The reference orientation (null rotation) is represented by $Q_0=1$, $Q_1 = Q_2 = Q_3 = 0$. Starting there, orientation may be integrated from equation (4). No singularities are encountered and no trigonometric functions need be evaluated.

Euler's four parameter method is as old as the method of Euler angles. It was not used by the aeronautical community until the need was created by the advent of automatic computers. It was then introduced to the industry by Robinson [5] and has since been gaining ground in flight simulation [6].

II QUATERNIONS

The four parameter method is generally known as the quaternion method. It should be pointed out that this is not merely a name. Rather, it refers to the fact that the four parameters Q_0, Q_1, Q_2, Q_3 together form a quaternion.

Quaternions were invented by Hamilton [7,8,9] about 140 years after Euler introduced his four parameter method. A quaternion may be considered the combination (sum) of a number Q_0 and a vector (Q_1, Q_2, Q_3). The quaternion may be written as

$$Q = Q_0 + Q_1\mathbf{i} + Q_2\mathbf{j} + Q_3\mathbf{k}, \tag{5}$$

where **i**, **j**, **k** are unit vectors along the coordinate directions. There is nothing out of the ordinary about formal addition of a number and a vector. The power of the method comes from the rules for multiplying quaternions. These may be derived from the multiplication table

$$\left|\begin{aligned} &\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1, \\ &\mathbf{ij} = -\mathbf{ji} = \mathbf{k}, \\ &\mathbf{jk} = -\mathbf{kj} = \mathbf{i}, \\ &\mathbf{ki} = -\mathbf{ik} = \mathbf{j}. \end{aligned}\right. \tag{6}$$

Quaternion multiplication is not commutative. However, there are no divisors of zero. A product of two quaternions cannot vanish unless at least one of them vanishes (all four components vanish). Any non zero quaternion can be inverted.

Numbers and vectors (in 3 dimensions) are special cases of quaternions. It is interesting to note that the quaternion product reduces to the ordinary product for these special cases. When each of two quaternions is a number, the quaternion product is the product of the two numbers. If one quaternion is a number and the other a vector, the quaternion product is the normal product of a number and a vector. When both quaternions are vectors, their quaternion product is a combination (sum) of the dot product and the cross product of the two vectors:

$$\mathbf{uv} = -\mathbf{u}.\mathbf{v} + \mathbf{u}\times\mathbf{v}. \tag{7}$$

The absolute value of a quaternion is defined as

$$|Q|^2 = Q_0^2 + Q_1^2 + Q_2^2 + Q_3^2. \tag{8}$$

This agrees with the definition of the absolute value of a number or of a vector, should the quaternion happen to be one of these. The absolute value may be expressed as

$$|Q|^2 = QQ^*, \tag{9}$$

where

$$Q^* = Q_0 - Q_1\mathbf{i} - Q_2\mathbf{j} - Q_3\mathbf{k}, \tag{10}$$

is called the conjugate† of Q. The inverse of a quaternion is $Q^*/|Q|$.

Rotations may be represented by quaternions and combined by quaternion multiplication. The rules for combining rotations are quite analogous for matrices and quaternions down to such details as multiplying on the left for rotations defined in earth axes and on the right for ones defined in body axes. Multiplication of quaterninons is less laborious than that of matrices. However the application of rotations to vectors is more straightforward in terms of rotation matrices.

The elegance and economy allowed by the quaternion formalism may be demonstrated by expressing the four equations (4) as a single quaternion equation

† The operation of taking the conjugate of a quaternion is reminiscent of complex numbers. As a matter of fact, complex numbers, too, are a special case of quaternions – the case where $Q_2 = Q_3 = 0$. The quaternion conjugate of a quaternion that happens to be a complex number is identical with the complex conjugate. However, we do not pursue this analogy here.

$$\dot{Q} = ½Q\omega. \tag{11}$$

The product of a quaternion and a vector in (7) is the quaternion product.

The benefit of reducing the four equations in (4) to the single extremely simple equation (11) is not limited to formal manipulations. It carries over into computer programming. This is demonstrated in the listing in Figure 1. The code is written in C++. A user defined class Q (for quaternion) has been created and endowed with quaternion properties including effect of arithmetic operators. Subroutine "step", shown in the listing performs one Euler integration step for the six degrees of freedom of a rigid body. Equation (11) is implemented verbatim. All vectors are defined in this class. 11 statements suffice to update 14 real variables (including time and the extra variable introduced by the use of quaternions) as well as perform additional manipulations. One of these is the application of the rotation induced by a quaternion q to transform the earth axes velocity Ve to body axes velocity Vb. The rotation procedure of one quaternion by another is defined by the one-liner in Figure 2.

```
void step(void)
{
  Mbb = Mb();
  t=t+dt;
  Xe = Xe + (Ve*dt);
  Ve = Ve + (Ae()*dt);
  q  = q + (q*Omegb)*(0.5*dt); q = q/abs(q);
  Vb = rot(~q,Ve);
  V  = abs(Ve);
  Omb.q1=Omb.q1 + (m1*(Omb.q2)*(Omb.q3)+(Mbb.q1)*li1)*dt;
  Omb.q2=Omb.q2 + (m2*(Omb.q3)*(Omb.q1)+(Mbb.q2)*li2)*dt;
  Omb.q3=Omb.q3 + (m3*(Omb.q1)*(Omb.q2)+(Mbb.q3)*li3)*dt;
};
```

Figure 1: Quaternions used in C++. The code performs one Euler integration step for a rigid airframe. The position (Xe), velocity (Ve), orientation (q) and angular velocity (Omb) are all quaternions.

```
Q rot(Q q, Q v) {return q*v*(~q);};
```

Figure 2: The subroutine by which a quaternion q induces a rotation in quaternion v.

Quaternions did not become dominant in mathematical physics, as Hamilton expected. Some of the reasons are that quaternions are limited to three dimensions, whereas physics graduated to four and more. Quaternions are also limited to scalars and vectors, whereas many applications require higher rank tensors. Still, when dealing with scalars and vectors in three dimensions, quaternions have a great deal to offer.

III THE PRICE

The use of quaternions to express orientation is not entirely without cost. The quaternion that expresses a rotation or orientation is a unit quaternion, i.e. it is subject to the condition

$$|Q| = 1. \tag{12}$$

Mathematically, the constraint is self preserving, but in the computer environment it must be enforced against truncation errors at every interation step. This may be done, as in Figure 1, by deviding by $|Q|$. However, in high precision applications where the constraint is virtually satisfied anyway, one may expand $1/|Q|$ around one and retain only terms only through first order. The constraint is then enforced by the code

```
q = q*(1.5 - 0.5*abs(q));                                    (13)
```

From the point of view of computer workload, this is a minor burden. Even so it is annoying that four parameters are required to specify three degrees of freedom. The impact on bandwidth, when the information has to be transmitted over a communications line may be significant. A nuisance is created when the quaternion method and, say, Euler angles need to coexist as alternative options.

Another source of complication is the fact that the quaternion representing an orientation is not unique. Rather, two distinct quaternions represent each orientation.

This paper remedies both the above objections by the introduction of Special Rotation Vectors (SRV).

IV ROTATION VECTORS

The problem of defining a continuous one to one mapping of orientations in three dimensions to a three dimensional manifold was solved by Wigner [10] in the context of the theory of atomic spectra. Euler had pointed out that any two orientations can be bridged by a single rotation defined by specifying the axis e and angle α. (e is a unit vector in the direction of the axis.) Wigner represented an orientation by a vector

$$\mathbf{w} \equiv (\alpha / \pi)\mathbf{e}, \tag{14}$$

constructed from the parameters e, α of the rotation that produces it from the reference orientation. The rotation is around the vector **w** in the positive screw sense. Wigner pointed out that the rotation angle α in (14) need range only from 0 to π. Orientations with α in the range π to 2π can be achieved by rotations of no more than π in the opposite direction, i.e. with the axis vector -e. The vectors **w** representing all orientations range on the closed unit sphere $|\mathbf{w}| \leq 1$. Opposite points on the surface of the sphere must be identified. (Rotating by π one way or the other results in the same orientation.) The unit sphere, with antipodal surface points identified, is the manifold of all orientations, with the wigner vectors **w** providing a continuous* one to one mapping.

Wigner's concern was only with the topological properties of his mapping. Its metrical properties are rather messy, as evidenced, e.g. by the form of the resulting kinematic equations. The desire for a more streamlined equation serves as motivator in the search for a variant scheme [11].

The topological properties of Wigner's mapping are preserved when (14) is replaced with

$$\mathbf{w} \equiv f(\alpha)\mathbf{e}, \tag{15}$$

where f is a continuous strictly monotonic function satisfying

$$f(0) = 0, \quad f(\pi) = 1. \tag{16}$$

The kinematic equations for the generalized Wigner rotation vectors of (15) are

$$d\mathbf{w}/dt = f'(\alpha)(\boldsymbol{\omega}.\mathbf{w})\mathbf{w}/w^2 \tag{17}$$

$$+ \tfrac{1}{2}w\cot(\tfrac{1}{2}\alpha)[\boldsymbol{\omega} - (\boldsymbol{\omega}.\mathbf{w})\mathbf{w}/w^2] + \tfrac{1}{2}\boldsymbol{\omega}\times\mathbf{w}.$$

This is quite messy for the general $f(\alpha)$ as well as for Wigner's original vectors. But it does simplify remarkably for

$$f(\alpha) = \sin(\tfrac{1}{2}\alpha). \tag{18}$$

This is hardly surprising. The vector

$$\mathbf{R} \equiv \sin(\tfrac{1}{2}\alpha)\mathbf{e} \tag{19}$$

together with

$$R_0 \equiv \cos(\tfrac{1}{2}\alpha) \tag{20}$$

are the solution of (4) (or of (11)) for a rotation around the axis e through an angle α. These are the components of the quaternion describing the orientation in question. The difference is that, not as in the context of quaternions, the angle α is restricted to

$$0 \leq \alpha \leq \pi. \tag{21}$$

In this range $\cos(\tfrac{1}{2}\alpha) \geq 0$. Therefore, Q_0 is entirely determined by **R** as

$$R_0 \equiv +(1 - \mathbf{R}^2)^{\frac{1}{2}}. \tag{22}$$

R_0 becomes a mere auxiliary quantity. The orientation is fully determined by the three component rotation vector **R**.

In quaternion practice the angle α is unrestricted. If instead of rotating by α, we rotate by $2\pi - \alpha$ the other way, the final orientation is the same, but the quaternion resulting from substituting these values in (19), (20) is not. Rather, all components change sign. Besides the quaternion $Q = R_0 + \mathbf{R}$, we obtain a second unit quaternion $Q' = -R_0 - \mathbf{R} = -Q$ also describing the same orientation and giving rise to the same rotation matrix (3). The vector part of the quaternion is no longer sufficient to determine the scalar part. It is no longer known whether the positive or negative square root should be used in solving the condition $|Q|=1$ for Q_0.

The angles of the two rotations add to 2π. Typically, one of them is a "short rotation", by less than π, the other a "long rotation", by more than π. By adopting Wigner's restriction (21), we embrace the short rotation and ignore the long rotation. Of the two quaternions that represent each orientation, we select the one with $Q_0 > 0$ and discard the one with $Q_0 < 0$. When $Q_0 = 0$, we identify the two rotation vectors. The mapping of orientations onto the remaining quaternions is one to one. Each of these quaternions is fully determined by its vector part - the Special Rotation Vector. The 2:1 mapping and the need for a fourth component are both eliminated†.

* The term "continuous mapping" implies a topology on the orientations as well as on the manifold into which they are mapped. Orientations form a metric space with the "distance" between two orientations being the angle of the (short, see below) rotation that bridges them. This metric is invariant under rotation of the axes and under change of the reference orientation. In terms of quaternions the "distance" between two rotations is δ_{12} = 2arccos(|number part of$(Q_1^* Q_2)$|). In terms of the Special Rotation Vectors introduced below, it is $\delta_{12} = 2\arccos(|\mathbf{R}_1.\mathbf{R}_2 + (1 - \mathbf{R}_1^2)(1 - \mathbf{R}_2^2)|)$.

† In the case of quaternions, Q_0 is determined by **Q** up to a sign. The three components of **Q** and one extra bit specifying the sign of Q_0 suffice to completely determine the orientation. In practice the treatment of odd bits is awkward. In any case, the information conveyed by this bit is whether the short or long rotation has been used to represent the given orientation.

The condition $Q_0 \geq 0$ is not self preserving under equation (4) (or (11)). It needs to be reinforced by the pseudo code

```
if($Q_0$ < 0)Q=-Q;                                    (23)
```

In mathematical notation the extraction of an SRV from a quaternion is defined by

$$\mathbf{R} = \begin{cases} \mathbf{Q}, \text{ if } Q_0 \geq 0, \\ \\ -\mathbf{Q}, \text{ if } Q_0 < 0. \end{cases} \tag{24}$$

Like all Wigner rotation vectors, SRV range over a boundary-less manifold consisting of the closed unit sphere with opposite surface points identified. An SRV can readily be completed to a quaternion by adding R_0 of (22).

The discussion above supposes that full quaternions are still used to construct rotation matrices through (3) or to integrate the time development using (4) or (11). In the case of (3) this is a moot point. Whether Q_0 is determined by (22) and then substituted in (3) or the substitution takes place in (3) directly is immaterial. However, an attempt to eliminate Q_0 from (4) by substituting $+(1 - \mathbf{R}^2)^{1/2}$ leads to a difficulty. The resulting equations develop a bifurcation on the surface of the unit sphere. The solution may be trapped on this surface.

The introduction of SRV [12,13] produces a one to one mapping of orientation‡ and permits expressing same by three parameters, while retaining all the computational advantages of the quaternion method. The price of propagating four components and enforcing a constraint is also retained.

In terms of computer code, the system of equations that maintains short rotation quaternion (whose vector part is the SRV) might be

```
Q q,Omb;                                              (25)
.....
q = q + (q*Omb)*(0.5*dt);
if(q.q0>0)q = q/abs(q); else q = -q/abs(q)
```

Omb is the (body axes) angular velocity. Both q and Omb belong to class (type) Q for quaternion.

The difference between quaternions and special rotation vectors may be illustrated by the simple case in which the angular velocity vector ω is a constant. Figure 3 shows the trace of the quaternion describing the resulting orientation. What is plotted is the vector part of the quaternion as it moves in

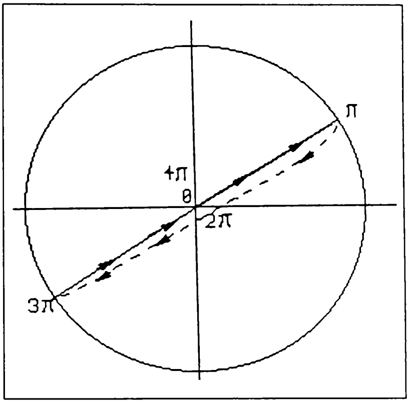


Figure 3: History of Quaternion for Constant ω.

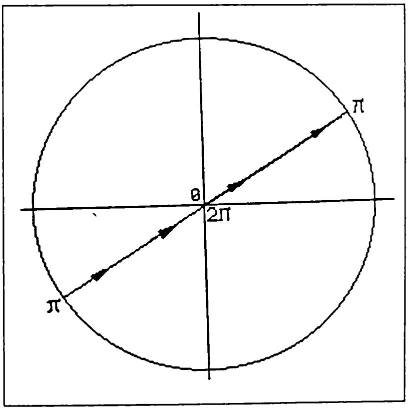


Figure 4: History of SRV for Constant ω.

‡ The unit quaternions are a representation of SU2 [12], the covering group of the rotation group. Rotations are the factor group of this group divided by the group containing the numbers 1, -1 (which represents the "rotations by 2π"). The elements of the factor group are pairs of quaternions which are the negative of each other.

the 3-dimensional unit sphere. The trace is shown solid when $Q_0>0$ and dashed when $Q_0<0$. We start in the reference orientation. The trace starts at the origin, with the initial value of the quaternion being 1, and the vector part vanishing. The trace moves out radially in the direction of ω and reaches the surface of the sphere after a rotation through an angle π. As the rotation continues, the trace moves back towards the origin. Q_0 crossed zero at the surface and is now negative. (The trace overwrites itself; it is shown offset for clarity.) The trace reaches the origin after a rotation of 2π. The value of the quaternion is now -1. The trace now continues towards the surface in the opposite direction. At the surface the direction of motion and the sign of Q_0 again reverse. The trace returns to the origin, and the value of the quaternion returns to 1 after a rotation by 4π.

Figure 4 shows the trace of the special rotation vector describing the same motion. It, too, starts from the origin and reaches the surface of the sphere after going through a rotation by π. However, in this case the point on the surface is identified with the opposite point. The trace returns into the sphere from this opposite point. The direction of motion of the trace (and the sign of the auxiliary R_0) never change. After a rotation of 2π, the trace returns to the origin with the values of all variables restored.

V SUMMARY

Special Rotation Vectors occupy the overlap between the formalisms of quaternions and of rotation vectors. They inherit the main virtues of each: the simple, singularity free, linear and quadratic equations of the Euler/Hamilton method; the continuity and uniqueness of Wigner rotation vectors. SRV allow a convenient option, not recognized previously, for the encoding, recording, and transmission of orientation.

REFERENCES

1. See A. Katz, "Topics in Dynamics", lecture notes, December 1991 for an exposition of the mathematical details.
2. Leonhard Euler, Novi Commentarii Academiae Petropolitanae **15**, pp13-15, 75-106 (1770).
3. Leonhard Euler, Novi Commentarii Academiae Petropolitanae **20**, p189 (1776).
4. Leonhard Euler, Novi Commentarii Academiae Petropolitanae **20**, p208 §6 et seq. (1776).
5. Alfred C. Robinson, "On the use of Quaternions in Simulation of Rigid Body Motion", Technical Report 58-17, US Air Force Wright Air Development Center, December 1958.
6. Brian Goldiez and Kuo-Chi Lin, Flight Simulation Technologies Conference, New Orleans, August 1991 published by AIAA, Washington 1991, p 271.
7. Sir William Rowan Hamilton, "On a new species of imaginary quantities connected with a theory of quaternions", Dublin Proc., **2**, no. 13, pp 424-434, November 1843.
8. Sir William Rowan Hamilton, Lectures on Quaternions, Dublin 1853.
9. Sir William Rowan Hamilton, Elements of Quaternions, Dublin 1865; second edition, Longman's, Green, and Co., London 1899.
10. Eugene P. Wigner, Group Theory and its Applications to the Quantum Theory of Atomic Spectra, Academic Press, New York 1959, p89, p151.
11. Amnon Katz, "Continuous One to One Mappings of Orientation in Three Dimensions", Sept. 1991, unpublished.
12. Amnon Katz "Special Rotation Vectors - A Means for Transmitting Quaternion Information in Three Components", Sept. 1991, to be published in Journal of Aircraft.
13. Amnon Katz "Special Rotation Vectors: Quaternions in Three Components", Position Paper, Sixth Workshop on Distributive Interactive Simulation, Orlando, March 1992, vol II, p 123.

AIAA-92-4170-CP

NETWORKED SIMULATION OF MULTIPLE AIRCRAFT USING SEMI-AUTOMATED FORCES

G. Robert Vrablik, Staff Engineer
Robert B. Calder, Engineer
BBN Systems and Technologies Division
Bolt Beranek and Newman, Inc.
Cambridge, MA

ABSTRACT

In this paper we focus on the development of a networked, semi-automated, fixed-wing and rotary-wing aircraft simulation. Semi-Automated Forces (SAF) is a system which simulates multiple (currently approximately 100) vehicles under the control of a single operator for use with networked simulations. This system was constructed to provide an extensive and realistic threat to manned ground vehicle and air vehicle simulators in a distributed simulation environment. The software which implemented the fixed-wing aircraft (FWA) and rotary-wing aircraft (RWA) simulation was built upon an existing SAF system, which operated in a ground vehicle environment. We will discuss the problems encountered and solutions produced during the development of this software, called AIRNET SAF. We will present a brief overview of the SAF system and the software's architecture. We will then discuss the command and control system, vehicle and unit behaviors, FWA dynamics model, and implementation of aircraft formation keeping.

SAF SYSTEM OVERVIEW

The Defense Advanced Research Projects Agency (DARPA), in cooperation with the U.S. Army, initiated the Simulation Network (SIMNET) program to develop the technology for large-scale networking of interactive combat system simulators.[1] This program produced a network of manned, interactive simulators for use in the training of U.S. Army ground and air vehicle crews. SIMNET has proven the applicability and feasibility of Distributed Interactive Simulation (DIS) to Army training, research, and development programs. The Semi-Automated Forces (SAF) is a system which simulates multiple vehicles on a single computer workstation in a DIS environment, under the control of a single operator. The SAF system was constructed to provide an extensive and realistic threat to manned ground vehicle and air vehicle simulators. SAF is essential for making research studies and large-scale DIS exercises economically viable by reducing the manpower and equipment required to support them. The quality of the SAF behavioral simulations is a key factor in determining the problems to which DIS can be applied. Various versions of SAF are currently installed and in use at many training and research sites, including the Airnet Advanced Simulation Warfighting and Development Complex at the U. S. Army Aviation Center at Ft. Rucker, Alabama.

The two main goals of the SAF system are to provide the simulation of a large number of computer-generated vehicles on a single computer workstation, and to produce behaviors in these vehicles which makes their actions on the simulated battlefield indistinguishable from the actions of manned simulators on the same battlefield. These two competing goals dictate not only the scope of intelligent behavior which can be represented by the SAF system, but also the software architecture of the system. The tradeoffs between the sophistication of the vehicle behaviors and the computational costs associated with these behaviors must be constantly evaluated, and a compromise must be reached which serves to satisfy both of the SAF system's main goals.

A typical assumption which is made about the SAF system is that as computer hardware inevitably becomes faster, a single workstation will be able to simulate larger numbers of SAF vehicles. However, as more computational power becomes available, there is also a desire to increase the complexity of the behaviors of the SAF vehicles in order to make them more realistic. This typically causes the SAF vehicle count to increase steadily over time, rather than dramatically as faster hardware platforms become available. For the current SAF system, it is recommended that no more than approximately 100 vehicles be simulated at one time. However, while simulating approximately 100 of its own vehicles (local vehicles), the current SAF system can interact with up to approximately 800 other simulated vehicles (remote vehicles), either manned or SAF, which are running on the same simulated battlefield at the same time.

The SAF system is divided into two components: the SAFstation and the SAFsim. The SAFstation provides the SAF user interface and a two-dimensional Plan View

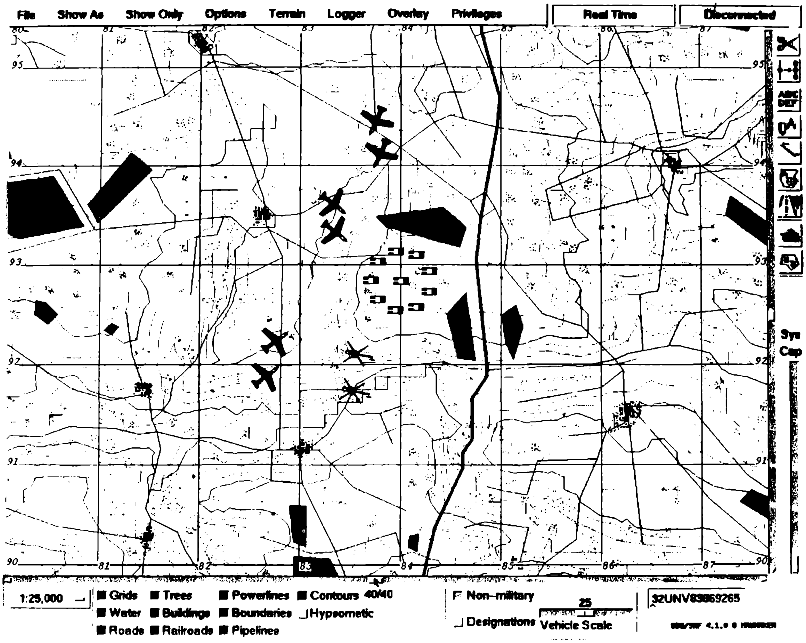


Figure 1: SAFstation Display

Display (PVD) of the simulated battlefield (see Figure 1). It is run on a single computer workstation, and is controlled by a human operator called the SAF commander. The SAFsim is used to simulate the vehicles which the SAF operator commands through the SAFstation. It is run on a single computer workstation, and communicates with the SAFstation through a shared Command and Control database via Ethernet.

The SAFstation is currently hosted on a MIPS UNIX workstation with software written in C and XWindows/Motif, but was previously hosted on a Symbolics workstation with software written in LISP. The SAFsim is currently hosted on a MIPS UNIX workstation with software written in C. Both the SAFstation and SAFsim software can easily be ported to run on any UNIX-based platform. Most of the current software has already been ported to run on Silicon Graphics Indigo workstations, Sun SPARC workstations, and SONY NEWS workstations.

The architectural description and software problems and solutions discussed throughout this paper all relate to the SAFsim software.

SAF SOFTWARE ARCHITECTURE

The SAF system's basic task is to simulate as many vehicles as it can in a realistic fashion. To accomplish this task, the system is designed to keep a list of the vehicles it is simulating, and to sequentially spend a small increment of processing time on each of these vehicles. The process of going through the list of vehicles once is called a "tick", and the process of handling a particular vehicle's requirements during a single pass through the local vehicle list is called "ticking" a vehicle.

The tasks that the system must perform during a vehicle's tick are based on what that vehicle is doing at the time. They can be a simple, single task, such as sending a vehicle's position out on the network when that vehicle isn't moving. Or they can be multiple, complex tasks such as performing complicated movements, tracking a target, firing a weapon, and performing multiple intervisibility and detection calculations on other potential targets. Since what a vehicle may be doing at any given moment can vary so dramatically, the amount of processor time which may be needed by a particular vehicle can not be allocated into a standard slice which covers any potential computational need. To do so would leave a significant amount of processor time unused. Instead, the vehicles essentially time share the system, each using what it needs at the moment, then passing the processor on to the next vehicle. This greatly increases the total possible vehicle "yield" for the system, for while vehicles in one area may be heavily engaged, those in another may not be consuming much processor time, in effect relinquishing their extra time to the other vehicles.

Though it does greatly increase the efficiency of the system, this architecture means that the programmer never knows how much time will pass between the ticks of a vehicle. This means that in designing modules for the system, events can not be anticipated. For instance, when programming actions that may occur in timed sequences, the programmer can not anticipate that the next event will occur during the next tick. The event must be handled after it occurs, because otherwise the vehicle might end up handling an anticipated event, get a shorter tick than expected, and find that it has handled an event that has not yet occurred.

The SAF system is object-based, treating each item that it must model in the simulated environment as an object. Objects are things that can change, move, or be changed, such as vehicles, missiles, and soldiers. Effects, such as explosions and muzzle bursts, are not objects. Each of the simulated objects has associated with it a set of specified parameters from configuration files which indicate to the simulation all of the capabilities for that particular type of object.

SAF vehicle tasks are implemented using state machines, which break each task up into successive stages, or states. By breaking each task into smaller pieces, it is easier to create modules to complete each separate part of a task. For instance, commanding a rotary wing aircraft to fly to a location is composed of the sequence of tasks of taking off, flying to the location, slowing down, and landing. The aspect of flying is composed of the non-state tasks of maintaining the proper direction, altitude, and speed. In addition, another task looks for potential targets or threats and can momentarily interrupt the task of flying to the location while the vehicle turns to engage, flee, or maneuver.

With this implementation technique, it is possible to create quite complex behaviors out of simple pieces. As more and more levels of these simple behaviors are added and combined, the overall behaviors of the vehicles become quite complex, and gain the appearance of intelligent control at the vehicle level. With traditional Artificial Intelligence techniques, much of the controlling behavior is built on top of large knowledge databases and decision making models which in the end hopefully achieve the realistic behaviors desired through the complexity of those models. This system instead seeks to create realistic behaviors by modeling these behaviors themselves, avoiding much of the computational overhead associated with the traditional technique. With this approach we can achieve a significant increase in the number of vehicles simulated, and since our primary goal is the behaviors themselves, a more realistic simulation.

SAF VEHICLE COMMAND AND CONTROL

Hierarchical Levels of Control

A SAF unit can be controlled at any level of its military hierarchy. Missions can be constructed for units of any size from battalions down to individual vehicles. While a higher-level unit is executing a mission, subordinate units can be issued a different mission. The higher-level unit will continue its current mission, adjusting for the lost subordinate units. This capability allows the SAF commander to exercise very precise control over a small number of vehicles while the remaining forces carry out their orders. At any time, the SAF commander can order subordinate units to terminate their mission and rejoin the higher-level unit.

Commanding SAF Units

The SAF commander can control SAF units in three ways. First, the SAF commander can plan predetermined missions for his units to execute by entering graphical control measures into an operations overlay. He can then task the units to execute these pre-planned missions by giving them Operations Orders (OPORDS). Second, if he finds that he must adjust his orders to compensate for changing battlefield conditions, he can edit the operations overlay and convey the changes to his units via a Fragmentary Order (FRAGO). Third, he can use a Tactical Emergency (TAC/E) command to interrupt the current missions of his forces and divert them to a new course of action. At any time, he can order a unit to resume its previous mission. These methods of control can be used separately or in any combination to allow the SAF commander to control the planned behavior of his units.

Combat Instruction Sets

A SAF unit's orders are represented by Combat Instruction Sets. (CISs). A CIS is a collection of parameters which describe the orders which a SAF unit is to obey in terms of movement, shooting, communication, and reaction. Table 1 shows an example CIS and its parameters, along with the range of acceptable values for each parameter, and the default value for each parameter. In the SAF system, there is a set of CISs for the enemy forces and a set of CISs for the friendly forces. This distinction is needed in order for the SAF system be able to represent the tactics of Blue as well as Red forces. The enemy and friendly CISs are grouped into different classes corresponding to the echelon types of the units which the SAF system can simulate: ground vehicles (tank, mechanized infantry, motorized rifle, ADA, dismounted infantry, etc.), rotary-wing aircraft, and fixed-wing aircraft. These classes are further divided into CISs corresponding to the different echelons in the military hierarchy which the SAF system can represent: battalions, companies, platoons, flights, sections, and individual vehicles. Each CIS corresponds to a common military procedure which is executed by a unit with the given tactics, echelon type, and echelon.

All orders which are issued by the SAF commander are either issued in terms of a CIS (in the case of OPORDS and FRAGOs), or are translated into a CIS by the SAF system software (in the case of TAC/Es). Therefore, a SAF unit will always be executing some CIS. This simplifies the software in that it only needs to know how to execute each of the different CISs in order to exhibit a robust set of tactically correct behaviors and missions.

As a SAF unit performs its mission on the battlefield, it will typically execute a number of different CISs. Sometimes it will execute a series of CISs, transitioning from one to the next until it has accomplished its planned objective. However, in most cases the SAF unit will need to perform certain CISs for only a fixed amount of time, and then return to a previous CIS which it was executing earlier. In order to accomplish this, each SAF unit has a CIS stack. The stack starts out with a single CIS on it, which is the CIS which the SAF unit is initially created with. When the SAF commander issues commands to the SAF unit, the unit will either transition to the new CIS by replacing the current CIS on the stack with the new CIS, or push the new CIS by placing it onto the stack on top of the current CIS. When a CIS which was pushed onto the stack expires and the unit is to resume to a previous CIS, the current CIS will be popped from the stack and the previous CIS will take over. The management of CISs and the stack is accomplished by implementing a set of CIS stacking/unstacking rules within the SAF software. The decision of whether to transition to or push the new CIS is based on whether the new CIS is one which gives the SAF unit a new set of orders (transition to this CIS), or is one which gives the SAF unit a set of orders which may expire at a later time in

Parameter Name	Range of Acceptable Values	Default Value
Name	Attack Ground Targets	Attack Ground Targets
Communicate	<any character string>	Performing Ground Attack
Speed	<any>	540.0 km/h
Altitude	<any>	50 meters
Formation	Fighting Wing, Line Abreast, Bearing, Box, Offset Box, Vic, Trail	Fighting Wing
Formation Scalefactor in X	0.1 - 10.0	1.0
Formation Scalefactor in Y	0.1 - 10.0	1.0
Attack Geometry	Split, Ninety-Ten, Direct, Trail	Split
Attack Entry	Pop-Up, Level, Standoff Pop-Up, Standoff Dive	Pop-Up
Attack Delivery	Low Altitude Dive, Medium Altitude Dive, Laydown, Strafe	Medium-Altitude-Dive
Fire Permission	Hold Fire, Fire At Will, Fire At Position	Fire At Position
Engagement Range	0 - 7000 meters	6000 meters
Enabled Weapons	<list of>: Maverick, Stinger, GAU-8, 500 lb. bomb	Maverick, Stinger, GAU-8, 500 lb. bomb
Target Location	<any X,Y or UTM coordinates>	N/A
Target Vehicle Priority List	<prioritized list of>: AAA, RWA, FWA, Tank, APC, C2, Artillery, Logistic	1=FWA,2=RWA,3=AAA, 4=Tank,5=APC,6=C2, 7=Artillery,8=Logistic
Start Time	Immediately, H-Hour +/- <any> minutes	Immediately
Enabled Reactions	Attack Complete	Attack Complete

Table 1: Blue FWA Flight "Attack Ground Targets" CIS

the mission (push this CIS). When a CIS is to be popped, the stacking/unstacking software makes a decision as to which CIS on the stack is the correct one to execute, and all CISs between this new CIS and the expired CIS on the stack are popped.

SAF VEHICLE BEHAVIORS

By utilizing the methods of controlling SAF units described above, the SAF commander has the capability to set up pre-planned missions and adjust those missions to react to changes which occur on the battlefield. This provides the SAF commander with the capability to continually modify his preplanned missions in order to give his units a better chance of successfully accomplishing their objectives. However, the SIMNET battlefield is a dynamic one, where all vehicles (SAF vehicles and manned vehicle simulators) taking part in the same exercise are interacting with one another. If these methods of controlling SAF vehicles were the only factors determining SAF vehicle behavior, the SAF commander would have to precisely monitor every situation which his forces encountered, make rapid decisions about how to react to each unexpected event, and modify each unit's mission quickly enough to effectively respond to the situation. This level of control for the SAF commander is difficult on a dynamic battlefield, even for a very small number of vehicles. It is necessary for the vehicles of the SAF system to possess some level of autonomy, not only to decrease the burden on the SAF commander, but also to give the SAF vehicles the appearance of possessing enough intelligence to handle certain situations on their own. Any automated behavior which the SAF system exhibits must contribute to making SAF vehicles and manned vehicle simulators indistinguishable on the simulated battlefield.

In addition to the control which the SAF commander exercises over his units, the behavior of SAF units is also determined by what we call reactive control. This reactive behavior is invoked automatically to react to conditions present on the battlefield, and is influenced by factors based on METT-T (Mission, Enemy, Troops, Terrain, and Time available). As a SAF unit carries out its mission, it continually assesses the battlefield situation around it by accumulating and assimilating METT-T information, and uses this processed information at two different behavioral levels. We call these two levels of behavior low-level behaviors and automated reactions, each of which is described below.

Low-Level Behaviors

The low-level behavior of SAF vehicles depends not only upon the mission which the SAF commander has ordered, but also upon METT-T factors such as the position of friendly forces, the position, heading, and strength of known enemy forces, the availability of concealment points from known enemy forces, the location of buildings, rivers, bridges, trees, treelines, and tree canopies, and the objectives of the current mission. A SAF vehicle will use these and other METT-T factors to decide how to execute its tactical operations. The use of METT-T factors is exhibited in a SAF ground vehicle's capability to avoid collisions with other vehicles on the battlefield, avoid collisions with obstacles on the battlefield, move tactically using concealment provided by trees and treelines, find routes through areas populated with obstacles such as buildings and trees, and find bridges and fording points when confronted with unfordable water. SAF air vehicles use METT-T factors in order to avoid collisions with other air vehicles and obstacles, and perform contour flight.

The low-level behaviors of a SAF unit are managed by a priority-based scheme. This scheme allows the low-level behaviors to factor into the manner in which a vehicle carries out its orders, and to work cooperatively with the current mission in order to achieve the goals of the SAF commander's desired mission. All low-level behaviors are executed after the mission's task state machine has run. Since the low-level behaviors are things which do not change the mission, they are executed just prior to deciding where a vehicle will move next. This restricts the effects of low-level behaviors to how the assigned orders are carried out. The behaviors are evaluated one at a time, and multiple behaviors can run in a single tick of the vehicle. The low-level behaviors with the most importance are evaluated last, ensuring that their recommendation for where the vehicle should move next will have the most bearing on where the vehicle actually goes. Some of the issues involved in creating the air vehicle low-level behaviors are discussed below.

The collision avoidance algorithm which was used for ground vehicles and RWA was inadequate for FWA because it assumed that the vehicles which were interacting had constant velocity. Since FWA spend much of their time turning, a new collision avoidance algorithm was developed for FWA which did not exclusively use the constant velocity assumption. The algorithm which was developed uses an algorithm based on "arc-line" collision avoidance.[2]

In previous versions of SAF, contour flight was not performed by air vehicles; instead they flew through any trees, treelines, tree canopies, buildings, or other obstacles which were in their path. For SAF air vehicles, a significant improvement in the flight characteristics was needed in order to make flight at low altitudes seem more realistic. Therefore, contour flight was implemented as a low-level behavior for both RWA and FWA. However, different algorithms are used for RWA and FWA, due to differences in vehicle performance characteristics and typical flight characteristics.

For SAF RWA, the algorithm for contour flight takes into account the fact that RWA can fly at altitudes as low as 3 meters AGL. In order to be able to fly over the terrain at high speeds at such a low altitude, the RWA needs to look ahead of its current position to determine if any obstacles, such as trees, treelines, or buildings, intersect with its immediate flight path. On each tick the vehicle retrieves terrain information from METT-T and evaluates the information within a rectangular area in front of its current flight path. The distance which the vehicle will look ahead varies based upon its current speed, but is always a constant number of seconds. If any obstacles in the vehicle's flight path are higher than its current altitude, the vehicle climbs to an altitude slightly higher than that of the obstacle. This algorithm is performed every tick, assuring that the vehicle will always fly at its commanded altitude or slightly above the altitude of the highest obstacle in its flight path.

In contrast to RWA, SAF FWA normally don't fly at altitudes lower than the highest obstacle on the terrain. Instead of having to avoid the obstacles, the FWA simply need to avoid the terrain due to the high speeds at which they fly. To accomplish this, the FWA computes the altitude at a point which is a set distance further down its flight path, by getting the altitude of the terrain at that point and adding its desired altitude. The intervening terrain is then sampled to determine if any of it intersects the path between the FWA and this point. If none does, then the FWA will fly directly toward this point. If any of the intervening terrain does intersect, then the flight path will be changed to pass over this intervening terrain. As with the formation keeping algorithms (to be discussed later), this calculation occurs every tick, so the point being flown to is always advancing ahead of the FWA, acting as a natural buffer so that the FWA's flight path will be smoother than the terrain it is following.

Automated Reactions

The low-level behaviors described above are automated behaviors which are typically executed by a single SAF vehicle. They are representative of actions which an individual vehicle crew member would execute in order to carry out his orders, and add the appearance of intelligence to the manner in which a single SAF vehicle carries out its assigned mission. However, they do not provide a SAF unit with tactically correct, contingent missions when it is confronted with a situation where it would not be in the best interest of the unit to continue its current mission. This capability is needed in the system to allow the SAF units to perform preplanned, tactically correct responses to common battlefield events and situations. These automated reactions are invoked if a certain tactical situation arises or a certain battlefield event occurs. These situations are evaluated, events recognized, and contingency missions executed not at the individual vehicle level, but at the unit level (i.e. section, flight, platoon, company).

The tactical situations and battlefield events which a SAF unit will react to are recognized as they occur, based on the METT-T information the unit has accumulated and battlefield events which it has learned about via the simulation network. The METT-T information is updated periodically, but the information from the simulation network is received asynchronously when battlefield events occur. Whenever a vehicle shoots at another vehicle, a mine or artillery shell bursts, a round impacts a vehicle, etc., that information is sent out on the simulation network. As SAF vehicles receive this information, they report it up the chain of command to their superior unit. The SAF units collect this information and when a unit recognizes a situation or observes an event which it should react to as specified in the objectives of its current mission, the planned mission behavior is automatically interrupted, and a new behavior is initiated which attempts to handle the situation or event in a tactically correct manner. In addition, the unit sends a report to the SAF commander indicating that it has recognized the situation or event and is reacting to it. When the situation no longer exists or the event has completed, the SAF unit normally resumes the mission which was interrupted. However, if the success of that mission was jeopardized by the situation or event which occurred, the SAF unit will report and await orders from the SAF commander.

Table 2 lists the tactical situations and battlefield events which SAF RWA units currently recognize, and the corresponding actions which are performed.

A large library of automated reactions relieves the SAF commander from having to specify every detail of a SAF unit's mission, while giving him the flexibility to override these automated reactions when issuing a mission. Because the reactions are data-driven from the CIS, the SAF commander can command his forces to respond to any combination of the automated reactions by simply modifying the mission CIS. By allowing him to specify how his units will react to the unpredictable situations and events which will occur, the SAF commander can customize scenarios to meet desired training goals.

A single automated reaction consists of a trigger-action pair. Each trigger corresponds to a function which will recognize the particular situation or event which that trigger is named for. Trigger functions are evaluated once per second at the unit level. Each action corresponds to both a CIS and a function; the CIS represents the contingency which will be activated, and the function determines how to activate that CIS in the context of the current mission. The CISs which correspond to actions are called reactionary CISs. They are typically set up to run

Situation or Event	Action
Enemy RWA Spotted with Hostile Intent	Jink, then based on fire permission and mission: Attack Enemy or Evade Enemy or Scramble Evade Enemy and Rendezvous
Under Attack from Enemy RWA	Jink, then based on fire permission and mission: Attack Enemy or Evade Enemy or Scramble Evade Enemy and Rendezvous
Enemy FWA Spotted with Hostile Intent	Jink, then Dive, then Fly at Enemy, then based on fire permission: Attack Enemy or Evade Enemy
Under Attack from Enemy FWA	Jink, then Dive, then Fly at Enemy, then based on fire permission: Attack Enemy or Evade Enemy
Enemy Anti-Aircraft Radar, Anti-Aircraft Artillery (AAA), or Surface-to-Air Missile (SAM) Launch Detected	Jink, then based on fire permission: Attack Enemy or Evade Enemy
All Targets in Area Destroyed	Resume Previous Mission

Table 2: Events and Situations which SAF RWA Units Recognize and Corresponding Actions

temporarily until the situation has been handled. The action function will cause either a push, pop, or transition to a new CIS on the CIS stack.

This trigger-action architectural implementation is flexible enough so that it can be used to signal the end of any CIS and cause any action to be performed, not just to implement the automated reactions. For example, we use this mechanism to implement CISs which execute for a specific time duration, and to implement SAF unit time-of-day coordination using H-Hour.

The automated reactions of a SAF unit are evaluated by a priority-based conflict resolution scheme. This means that the trigger functions are evaluated in the order in which they are listed in the CIS, and once any trigger function evaluates to TRUE, its action function is applied to its reactionary CIS. Therefore, if two triggers were to become TRUE at the same time, the one which appears first in the CIS will have its action executed. If, upon resuming to the original mission CIS after reacting to the first trigger function, the second trigger function still holds true, its action will be executed. The current implementation does not handle reactions within reactions.

FWA DYNAMICS MODEL

The fidelity of the FWA vehicle dynamics model that the SAF is using is not as high as it could possibly be. The model currently implemented is complex enough to handle any maneuver required in a realistic fashion without spending a significant amount of processor time doing complex calculations to accomplish an exacting adherence to the laws of physics governing an aircraft. This level of fidelity was chosen for the system based on the need for a maximal number of vehicles per SAFsim. A more accurate model could have been used, but it would have significantly reduced the maximum possible number of vehicles simulated. For most users of the current system this would have been an unacceptable exchange, since from more than a few meters away the level of fidelity of the model is not apparent.

This kind of decision is constantly made in the process of developing the SAF system. A balancing point has to be made between true fidelity and the amount of processor time it will take to achieve that fidelity, and therefore how many vehicles the system will end up being able to simulate. Generally, the level of fidelity is chosen to satisfy the needs of all but the most exacting user in a particular situation.

FORMATION KEEPING

The SAF system's original formation keeping algorithm used for air vehicles was the same as the simple ground vehicle model we had been using. It was based on a vehicle attempting to stay in its formation position relative to the position and direction of the vehicle immediately ahead or beside it in the formation. At first glance, it seemed like it would work fine, but in practice, it didn't. There turned out to be a multitude of problems: FWA were overshooting their positions in the formation when approaching from behind, were oscillating around their correct positions, and were flying off in the wrong direction when the lead vehicle turned.

The approach to solving these problems was to first break the problem up into two parts: following the leader in formation in straight and level flight, and performing formation turns.

Straight and Level Flight

The problem with straight and level flight seemed to be that the algorithm was too rigid. A vehicle ended up adjusting itself too much when trying to get into or stay in formation. It would get into position, but when it got there it wouldn't be oriented correctly or be going the proper velocity to be able to maintain the position in formation. The vehicle would then end up passing the point in the formation it needed to be in, and would have to start the whole process of trying to get into position all over again. Another problem was that some vehicles were following other vehicles, which were in turn following the lead vehicle, setting up a "chain" of formation keeping. This caused a magnification of any oscillation or movement by the lead vehicle in the movement of the other vehicles following in the chain.

The solution for this last problem was to establish each vehicle's position in the formation in relation to the one leading vehicle of the whole formation, instead of having a chain of vehicles. The position in relation to the leading vehicle is referred to as the vehicle's formation offset. Since each vehicle would then be following the leader directly, the error for each vehicle from the position it should be in at any moment will only be its own. It will no longer include the error in position of any other vehicle as with the chained formation keeping method.

Next was needed some method of "relaxing" the formation keeping code so vehicles would get into place and stay there without excessive problems. The old algorithm can be thought of as a steel bar affixed to the tail end of the leading vehicle, with the other end of the bar being the position the following vehicle needed to be in. If the lead vehicle turned a bit to the left, the end of the bar actually moved to the right, causing the following vehicle to head in the wrong direction (see Figure 2).

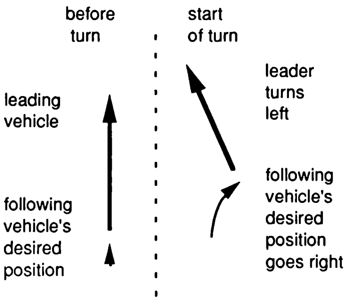

before turn
start of turn
leading vehicle
leader turns left
following vehicle's desired position
following vehicle's desired position goes right

Figure 2

The first thought was to simply relax this by doing the computational equivalent of putting a hinge on the tail of the leading vehicle. This would have taken care of the following vehicle turning in the wrong direction, but not the oscillation problems. A buffer was still needed. What we eventually implemented was computationally inexpensive (always important with the SAF system), and seemed to inherently solve most of the problems of straight and level flight. The following vehicle now keeps position in the formation using two different pieces of information, the intended direction of travel of the lead vehicle (instead of its current direction of travel), and the anticipated position of the leading vehicle projected two seconds along its intended direction of travel (instead of its current location). For example, if the lead vehicle is following a route, and has just started to pass a point on the route involving a turn, then its intended direction is the vector from its current location to the next point on the route, not the direction it may currently be facing, or turning away from. The position which the vehicle doing formation keeping will try to head toward will be the formation offset (the formation position in relation to the leading vehicle) along the desired direction vector (the intended direction of travel) of the leading vehicle, from the point on the route which is two seconds along the desired direction vector at the leading vehicle's intended velocity.

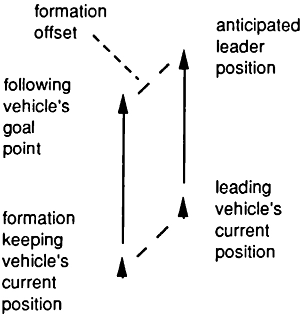

formation offset
anticipated leader position
following vehicle's goal point
leading vehicle's current position
formation keeping vehicle's current position

Figure 3

Since the following vehicle is always trying to be in the correct position two seconds from the current moment, there is an automatic buffer on vehicles approaching the correct formation position. Since the average tick is much less than two seconds, when the vehicle is next ticked and the new point to head toward is chosen, the vehicle will only have traveled a fraction of the distance to the original point. In effect, the distance the vehicle is closing to get into position is always being reduced, with momentum making up the difference. As the vehicle gets closer to the relative position it should be in in the formation, the velocity it will be flying to get into the proper position using the two seconds method slowly drops down until it has the same distance to go in that two seconds as the leading vehicle, and therefore will fly at the same velocity as the leading vehicle (see Figure 3). The other oddity about this method is that the vehicle is never actually using the numbers it is calculating to fly in its formation position. It is actually always trying to head toward the correct position it needs to be in two seconds ahead, even though the vehicle does end up flying in the proper position in relation to the leading vehicle.

Formation Turns

The second part of the formation keeping problem was making vehicles turn while in formation, and this turned out to be a harder problem to fix. The old model had the vehicles on the inside of a turn having to slow down and turn sharper, and those on the outside having to rush around the outside, traveling twice as far and twice as fast. While this method works, and is still used for shallow turns, in sharper turns significant problems arose. The model also didn't take into account the fact that in standard practice FWA switch sides of the formation during a hard turn.

The solution for formation keeping in a hard turn turned out to be, oddly enough, not to do formation keeping in turns. The vehicles of the formation were to fly at the same velocity to different points in space in relation to the lead vehicle's anticipated turn point, and to turn when they got there, then start formation keeping off of the lead vehicle if it had finished its turn. These points are chosen in such a way that each vehicle travels the same distance at the same speed, so therefore the vehicles should finish the turn in formation.

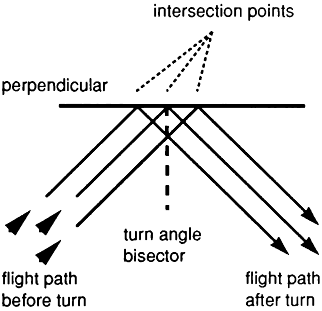


Figure 4

The point for the turn for each vehicle is chosen by bisecting the angle of the turn of the route the lead vehicle is following, and drawing the perpendicular to that vector at the point of the leader's turn. The path of each vehicle in formation entering the turn is then extended until it intersects this perpendicular. This intersection point is the point of the turn for that particular vehicle (see Figure 4). A simpler way of describing this is as a line of ping-pong balls rolling along next to each other in a line and bouncing off of a wall that lies in the path of travel at 45 degrees. By combining the formation keeping on the straight sections of routes and the independent flying techniques in the turns, we end up with what appears to be quite complex behavior.

Some other simple additions needed to be made for the model to work well. The velocity of the vehicles needed to be decreased when they were turning more than a fixed number of degrees off of their current heading, so they wouldn't speed up if they happened to be heading in the opposite direction from the one they wanted. Using the previously defined model would actually make them fly faster in the wrong direction. This acceleration would in turn slow the change of direction (max Gs in turn and speed slow the turning rate), and move the vehicle even farther out of position. The shallow turn method was causing the formation to narrow by the end of the turn. This was solved by having vehicles in a shallow turn go to intermediate points in the turn, determined by their formation positions relative to the leader. These points are the formation positions rotated half the total angle of the turn, and serve to pull the vehicles more cleanly through a shallow turn, since the act of turning is split into two smaller parts.

FUTURE WORK

There are a variety of programs currently underway which are contributing to the advancement of the SAF system. These programs will significantly increase the capabilities of the current system in various areas.

Work has begun on the DARPA WISSARD (What If Simulation System for Aviation Research and Development) program to use SAF and other systems to develop an advanced, synthetic, virtual environment for manned, high performance aircraft combat simulation to support advanced tactical training, doctrine development, and evaluation. This work will add many capabilities to the SAF FWA, including beyond visual range air to air combat tactics, more advanced radar and communications models, more reactive behaviors, and EW/ECM (Electronic Warfare/ Electronic Counter Measures) simulation.

This WISSARD work is also enabling us to proceed with work to develop SAF into an extensible, layered set of software modules with well defined and documented program/library interfaces. Modular SAF (ModSAF), will allow rapid development and testing of new DIS agents, organizations, and tools. Users will be able to modify or replace any of its modules to suit their experiments or use the modules in their systems for custom applications. Users perhaps finding a need for a FWA utilizing a more exacting vehicle dynamics model than the one supplied with the current SAF system would be able to "plug in" their own model with the level of fidelity they desired. Since this application would be specific to this particular researcher, the rest of the user community would not have to suffer the reduction in the maximum number of vehicles simulated that this change would entail.

Another important advancement being explored in SAF technology is the ability to simulate SAF vehicles at varied levels of complexity, or granularity, in order to increase the number of vehicles which can be simultaneously simulated on a single computer workstation. This will serve to address the problem of tradeoffs between vehicle behavior complexity and the number of SAF vehicles which can be simulated on a single computer workstation, since the complexity of the SAF vehicle behaviors may not need to be as great in a large-scale exercise such as a war game.

Seamless simulation will investigate the possibility of interfacing real-world platforms with the virtual DIS world. FWA fighter aircraft will engage in beyond visual range combat with SAF generated aircraft in a DIS environment. SAF aircraft will send information indicating their position and state over the network which will stimulate the radar of the real aircraft. The position of the real aircraft will similarly be sent to project it into the simulated battlefield to interact with SAF. SAF will be one of the key elements in achieving this goal.

CONCLUSIONS

The methods presented here have generally performed quite well. The SAF system is being delivered to more places for various different uses as time goes on. It is no longer just the training tool it was originally developed to be. It has been used as a tool to build a historical recreation of the battle of 73 Easting from the Gulf War, simulating each event of the battle in fine detail. It is being used as a tool to help develop new weapon systems, and as an evaluation tool to evaluate both potential weapon systems, and new combat and search tactics.

The SAF system will continue to evolve, and its evolution will be driven by the needs of future users of the system. Each member of the growing user community needs more and different features to directly apply SAF to their application. With the advent of ModSAF, each user will be able to take the standard SAF product, and plug in their own specific functional modules in place of the standard modules supplied. This will allow them to be able to achieve their specific training, testing, or developmental needs. This system will save money and time for various projects, since development will only involve the few specific modules needed, instead of having to develop an entire new system. It will also have the added benefit of allowing easy interfacing with other testing and training development groups which are also using ModSAF and networked simulation.

ACKNOWLEDGEMENTS

This work was supported in part by DARPA SIMNET programs under contract number MDA972-90-C-0061, and in part by USA ARMY ETL contracts for the ODIN program, contract numbers DACA76091-C-0005 and DACA76091-C-0006.

REFERENCES

1 Pope, A.R., and R.L. Schaffer, "The SIMNET Network and Protocols," BBN Report No. 7627, BBN Systems and Technologies, Cambridge, MA, June 1991.

2 Brock, D.L., D.J. Montana, and A.Z. Ceranowicz, "Coordination and Control of Multiple Autonomous Vehicles," Proceedings of the IEEE International Conference on Robotics and Automation, Nice, France, May 1992, p. 2725.

AIAA-92-4171-CP

Dead Reckoning for Aircraft in Distributed Interactive Simulation

Suresh Goel, Ph. D.
Kenneth D. Morris

Northrop Corporation
Flight Simulation Laboratory
1 Northrop Avenue, T272/64
Hawthorne, CA 90250
(310) 332-6776

ABSTRACT

Networking large numbers of simulators over long distances is a very promising concept that has been demonstrated recently with SIMNET. This concept seems to work well with tanks and other slow moving vehicles. However, a question exists whether or not it will work with fast moving highly maneuverable fighter aircraft. Distributed Interactive Simulation (DIS) has taken on the task of making this happen. A technique, used in SIMNET and to be expanded in DIS, called dead reckoning, reduces the number of network updates required of each participant. This makes better use of the available bandwidth. This paper discusses the implementation and evaluation of a number of extrapolation algorithms. Comparisons are made between first and second order extrapolation and various methods of performing second order extrapolation delays caused by the large distances involved are compensated and the effect is studied. The large jumps caused by low frequency updating are smoothed by several algorithms and the results compared. Overall, very good results were obtained using second order extrapolation by simple Euler algorithms and smoothing the results.

INTRODUCTION

Along with the relaxed post cold war world atmosphere and the fall of the Berlin wall comes the challenge to remain prepared for all contingencies. This challenge has become more difficult as the military's ability to train for wartime tasks has been drastically restricted. For example, from an ecological point of view, low level training flights have been essentially banned in Europe. Public outcry has also restricted such activities in the United States for years. None the less, this training is vital as demonstrated by the success in Desert Storm. Therefore, in order to continue training as well as expand the scope of training, technological advances in the field of simulation must be provided. Currently, simulation training is conducted individually or in very small groups. This method lacks the interaction of larger forces and prevents the participants from experiencing the "fog of war" which complicates operations in actual combat.

During the 80's, DARPA in partnership with the U. S. Army began working on a solution to this problem. The SIMNET program which endorses the concept of a large number of simulators interoperating on a long distance computer network was developed and demonstrated.

The primary focus of SIMNET is slow moving ground vehicles. In order to enable this capability to include fast moving vehicles such as fighter aircraft and missiles, a new concept needed to be explored. This concept, called Distributed Interactive Simulation (DIS), is being built based upon the foundation pioneered by SIMNET.

The applicability of DIS is expanding in a multi dimensional fashion. Originally, the main focus was training and acquisition. Providing an electronic battlefield wherein opposing commanders and their forces could practice tactics also provides an arena in which new weapon technology could be validated. Then, validation could occur in the environment for which the technology was being designed prior to production. DIS enters the operational arena with advances such as the electronic sand box of DARPA's project ODIN in which real time battle monitoring is added to tactical rehearsal. Conceptual advances aimed at solving the problems associated with using our new technology in actual combat can also be pursued with DIS. This will allow technology to be defined which solves real world problems. As can be seen, DIS now has been expanded to include training, acquisition, operations, and technology.

As the DIS concept matures, many different simulation platforms of various levels of fidelity will be operating with one-another. Early in the development, tanks and other ground vehicles were the only participants. To this, nap-of-earth aircraft such as helicopters and forward air controllers were added. Then close air support entered the picture with aggressor aircraft to defend against the air-to-ground threats. These aggressors were engaged air-to-air to counter their effect. Naval vessels became involved by launching strike aircraft, landing forces, and ship borne bombardment. There is also aerial threat to the fleet that must be countered. Sub-surface interaction must be considered as submarines engage both surface and sub-surface targets. Orbital systems can become involved when we start addressing threat and damage assessment and begin entering intelligence utilization areas.

STATEMENT OF THE PROBLEM

Simulation devices will be of varied costs, capabilities, fidelity, and operational frame rates. DIS must be able to handle them all. With the expansion of DIS participants comes a similar expansion of simulated environments. The original SIMNET operated in a visual only mode. When aircraft, surface vessels, and subsurface vessels enter the simulated battle, the environment must be expanded. In addition to the visual mode, IR must be used. Radar becomes important as well as the optic, acoustic, electronic, and nuclear spectrum.

When the additional number of participants along with their expanded environmental requirements are considered, the communication requirements between simulation entities will grow as well. Existing and future high fidelity simulators will be mixed with low cost, low fidelity devices. The high fidelity devices generally operate at frequencies of twenty to one hundred frames per second while the low cost devices operate from one to fifteen frames per second. When all of these devices communicate through a long distance network, a technique must be used to reduce the number of Protocol Data Units (PDUs) each participant produces.

DIS implies large distances between simulation sites. This, in turn, causes a delay from the time a message is sent to the time that it is received and used by a distant entity. Therefore, the approach is to use dead reckoning to compensate for this delay and reduce the number of messages on the network.

Dead reckoning is a navigational concept used throughout history, in sea faring, in this century for aircraft navigation, and most recently in simulation. The basic premise is that a certain heading is steered for at a given time in order to arrive at a preplanned location. Dead reckoning works reasonably well in navigation. In simulation, the heading and rates at a given time are used to produce new positions by extrapolation.

Simulations are typically divided into a given number of program loops, known as frames. During each frame the position and attitude of an entity, for example, an aircraft, are computed using the equations of motion applicable to that body. Thus, a flight path is produced by sampling pilot input at frame rate and applying the appropriate computations. With dead reckoning, a parallel entity called a world coordinate set is used. Its flight path is determined by sampling the simulators position, attitude and rates at specified times and then extrapolating these to produce new positions and attitudes each frame.

A threshold is used to determine when updates occur. Essentially, the simulation is compared with the World Coordinate Set (WCS). When a predetermined difference exists from the threshold, an update occurs. This update corrects the WCS to the simulator conditions and the extrapolation continues. Updated position and attitude is sent to the network and all participants. Each participant maintains an extrapolated WCS of themselves and of each of the other participants. These are called Remote Vehicle Approximations (RVA). Each simulation view of the world is through RVA's which are updated by each host participant as needed.

Each time an update is made, the sending computer places a time stamp in the message indicating when the data was valid. This information is used by receiving simulators to compute, by extrapolation, the current position of the sending entity. Depending upon the maneuver being executed, the velocity of the aircraft, and the delay incurred, the updated position can be fairly close or somewhat far from the actual simulated position. In any case, with long delays, the corrected position will be closer than an uncompensated update.

The extrapolation used in dead reckoning can be performed using many different algorithms. SIMNET uses a first order extrapolation for position only. First order extrapolations produce straight line approximations. A second order extrapolation would produce curved paths which should more closely follow the original flight path. Second order extrapolation uses acceleration with the velocities used in a first order extrapolation. There are various different algorithms which can be used to do these extrapolations. Generally they use more samples of past positions and/or accelerations and can be quite complex. Network update rate is affected by the algorithm chosen. If the extrapolated path is able to follow the actual flight path more closely, then there will be fewer updates for a given threshold. Of course, if the threshold is reduced, the update rate increases. The main consideration with threshold and choosing an algorithm is to minimize the network traffic.

When considering delay compensation, higher update rates will not improve accuracy. When an update is received, it is immediately extrapolated the proper number of frames for the delay without regard to any threshold, which is not available at the remote site in any case. If the delay is long, the corrected position will have an induced error. It is important that the extrapolated path follow the actual path as closely as possible.

NETWORK SIMULATION AND ALGORITHM DEVELOPMENT TOOL

The Flight Simulation Laboratory (FSL) at Northrop lends itself to the study and implementation of various DIS concepts. The FSL Multiple Engagement Simulation (MES) provides between twenty participants (13 manned) and uses eight Encore 32/9780 series minicomputers, GE Compuscene III and IV Image Generators, and several graphics and array processors. A distributed network simulation version of the MES software was developed to study DIS concepts. The distributed network simulation (dead reckoning models) allowed network-like communications between the MES and a network participant running on an adjacent computer. The capability to induce various delays caused by long distances were incorporated into the network simulation.

The network simulation is an algorithm development tool that was created for studying the dead reckoning phenomenology. It was designed so that various changes to the software could easily be made on-line and off-line. A partial list of these features include on-line selection of different dead reckoning and smoothing algorithms as well as the capability to change update thresholds, constants in dead reckoning equations, and transport delay parameters. Off-line features included entering new dead reckoning and smoothing algorithms. Since laboratory setup did not have a link to any remote site, various off site networks were simulated. This simulation had one flight path and four different dead reckoned participants with different characteristics. These characteristics were controlled by using data files, on-line real-time changes, and touch screen inputs.

The simulation uses several touch screen input switches for control purposes. The Coordinate Set 1 switch provided the choice of selecting the actual flight path flown or selecting a previously recorded flight path. Each flight path could be replayed in either of the two modes, Stick or Position/Attitude. The Coordinate Set 2 - Coordinate Set 5 switches allowed the selection of up to four different dead reckoning models with different characteristics at the same time based on the same flight path. The simulation allowed the capability to setup over six million cases of which any four cases could be viewed through the Compuscene at the same time. Five switches for flightpath 1 and Dead Reckoning setup 1-4, were used for visual purposes only. This removed the visual model for the simulation while the coordinate set data continued. The grey aircraft represented the actual flight path and the brown, yellow, green, and red aircraft represented four different dead reckoning models.

The following 12 maneuvers were the basis of this study. The first eleven were recorded using the Northrop flight controls simulator. The main purpose for these is to provide a means for testing the effect of specific attitude rates and maneuvers on the overall update rate. The last one was provided by Lt. Col. David Greschke of 405 TTs/FAC aces at Luke AFB, AZ. This was a 2 v 1 engagement flown by experienced TAC pilots using the SAAC simulator at Luke AFB. This is the maneuver set:

1. Straight and level flight at constant velocity
2. Straight and level flight at varying velocities
3. Vertical "S" maneuver
4. High G turn at low velocity
5. High G turn at medium velocity
6. High G turn at high velocity
7. Loop at 45 degree role angle
8. Upright loop
9. Horizontal "S" turns (reversals)
10. Barrel roll
11. Random jinking maneuvers
12. Luke AFB flight path - 2 versus 1 line abreast

Maneuver number 9, horizontal "S" turns was particularly useful in watching the effects of delay and smoothing. This maneuver demonstrated that the role axis of fighter aircraft has the highest frequency response and therefore the highest possibility for error during extrapolation.

Synchronized updates were provided periodically for a specified number of frames based on touch screen inputs. These were used to test constant update rates. (i.e. 2 per second, 4 per second, etc.). All four dead reckoning models had a choice of different position and attitude thresholds. These thresholds are initially read from a data file and could be changed in real time using the real-time debugger tool.

There were two modes of recording any flight path flown. The first was recording the joystick's rudder, pitch, and roll deflections. The second was recording position X, Y, Z, velocities $\dot{X}$, $\dot{Y}$, $\dot{Z}$, accelerations $\ddot{X}$, $\ddot{Y}$, $\ddot{Z}$, attitude ϕ, θ, ψ, and their rates P, Q, R. Both methods repeated the actual flight path exactly when replayed. The touch screen switches STK for stick inputs and P/A for position and attitude were used to select each mode. For replay, the same two modes were available with several other features in the P/A mode. The features for replay mode were:

Q	Scenario start point selection
REW	Rewind scenario to beginning
REV	Play frames in reverse mode (backward)
STOP	Temporary pause
FWD	Play frames forward (default)
SNGL	Plays one frame at a time
SLOW	Slows frames increment (pressing this switch N times slows moving in either direction by 1/N+1 times)
NORM	Brings speed back to normal (20HZ, default mode)
FAST	Speeds up frames increment (pressing this switch N times speeds up to N+1 times in either direction)
1FRM	Used for visual purposes only, when ON the Compuscene IV does not receive the rates
GO	Start flying the scenario

Five independent flight paths (Coordinate Sets) were available. Each flight path could be selected by turning switches labeled CS1,...,CS5 OFF/ON. Since the same pointer was used in all recorded flight path data, the same replay features applied to all coordinate sets.

There were two (OFF/ON) modes for Big Brother View (BBV). The first mode BBV ON provided a view of the aircraft from the outside as controlled by the joystick. SEL1,...,SEL5 were mutually exclusive switch selections for one of the five coordinate sets (aircraft) supported. The joystick provided control for moving around each aircraft selected. The second mode, BBV OFF, provided a view from inside the cockpit of the aircraft selected. Additional switches provided a fixed position relative to the selected flight path (about 50 feet back and on the longitudinal axis), fixed the positions of all replayed flight paths to the first coordinate set to allow examination of attitude differences, and an in trail position that flies the selected flight path at a specified number of frames behind to provide a "Separate" pilot view, while keeping the subject aircraft within the view window.

To measure the performance and effectiveness of different algorithms, an off-line task was created which gave the update rates for four different dead reckoning models, average difference (deltas) for various coordinate sets along various axes for all twelve maneuvers, both individually and collectively. The items measured for any of the four setups were:

Update Rate - The number of updates per second, length of the scenario, and the total number of updates.

Position errors for Dead Reckoning Model - The average distance between actual flight path and the dead reckoning models along all axes.

Position errors for Visual Model - The average difference between the actual flight path and the visual coordinate sets (including delay, compensation and smoothing) along all axes.

These results for various extrapolation techniques are summarized in Tables 1 through 3.

To get the effect of changes made to a particular algorithm, a program was created to display the values of various parameters for on-line analysis of the behavior of the algorithms. The visual displays on the Compuscene IV and the statistical display gave instant verification and credibility to algorithm development and implementation.

AVERAGE UPDATES/SEC AND ERRORS FOR 12 SCENARIOS					
	UPDATES PER SEC	POSITION ERROR SMOOTHING		ATTITUDE ERROR SMOOTHING	
		OFF	ON	OFF	ON
	FIRST ORDER EXTRAPOLATION				
	2.422	3.16	22.30	0.20	0.20
	SECOND ORDER EXTRAPOLATIONS				
1	1.149	2.46	6.67	0.42	0.42
2	1.170	2.79	6.49	0.41	0.41
3	1.174	2.68	6.43	0.40	0.40
4	1.167	2.61	6.45	0.41	0.41
5	1.180	2.79	6.42	0.40	0.40
6	1.174	2.72	6.43	0.41	0.41
7	1.173	2.64	6.46	0.40	0.40
8	1.177	2.84	6.41	0.40	0.40
9	1.171	2.76	6.43	0.41	0.41
10	1.169	2.70	6.45	0.41	0.41
11	1.173	2.80	6.45	0.40	0.40
12	1.129	3.10	7.35	0.43	0.43
13	1.153	3.25	7.10	0.41	0.41
14	1.152	3.18	7.16	0.42	0.42
15	1.145	3.15	7.13	0.42	0.42

Average Update Rate/Sec and Errors for all 12 Scenarios
Using Various Extrapolation techniques.
The Update Envelope Used Was 9.1 Feet and 3 Degrees.

TABLE 1

OBJECTIVE

The objective of the study was to develop algorithms optimum tradeoff between minimum network traffic (i.e., minimum number of updates) and minimum position and attitude errors for a Remote Vehicle Approximation (RVA) and its high fidelity vehicle as used in real time DIS. The objective was compounded by the different frame rates of participating simulators.

APPROACH

It is useful to visualize, monitor, measure, and analyze the impacts of algorithm development for dead reckoning extrapolations, delay compensations, smoothing, and network traffic. Software tools were developed to be used in real time simulation and off-line analysis. One of the most useful tools developed is known as "Big Brother". This is a viewpoint control capability similar to the SIMNET Stealth Vehicle (a network entity that can be attached to any vehicle or flown free). To be consistent with flight characteristics for algorithm evaluations, record replay capabilities were developed. For data monitoring and evaluation, real time terminal displays and off-line printouts and graphs for statistical analysis were produced.

Euler's method is the most common technique for extrapolating new positions and attitudes. This method is a one-step method. Multi-step methods based on numerical quadrature formulas were studied. Some of these techniques were modified to adjust for the difference between the extrapolated and new updated parameter values, subject to some limits and the number of frames between updates. The experiments confirmed that these modifications to the standard techniques worked very well during maneuvering flight.

Delays of up to one second were implemented using a circular buffer. This allowed a different amount of delay from 0 to 1 second in increments of 50 msec (simulation frame time) for different dead reckoning models. The same extrapolation techniques were used for both dead reckoning and delay compensation. Using the time stamp which was provided in the transmitted Appearance Protocol Data Unit, the received information was extrapolated to the current position and attitude during one frame time and thereafter updated at the frame rate.

Five different smoothing techniques were selected for evaluation. Four tables were created to store the rate of adjustment at each step for the differences between the extrapolated position and the visual coordinate sets. The number of steps used and the amount of compensation applied were subject to the adjustment needed and the associated table. Flexibility of upscaling/downscaling of adjustment was provided using a touchscreen display. A running total of the adjustments applied was kept to avoid excessive adjustment.

EXTRAPOLATION, DELAY COMPENSATION, AND SMOOTHING ALGORITHMS

The DIS protocol refers to Euler's method for extrapolation of new positions. This method is a one-step method since the information involves approximation from only one previous point (step). Methods using the approximation at more than one previous point to determine the approximation of the next point are called multi-step methods or methods based on numerical quadrature formulas. These quadrature formulas have two categories, open method or closed method. Only quadrature multi-step and multi-order (QO.q.m) methods were studied. QO.q.m means quadrature open method with q-steps and of order m. All these methods are explained in most of the numerical analysis text books. All of the dead reckoning used models had a choice of one of the five velocity extrapolation techniques and one of the sixteen position extrapolation techniques as described below. A description of the notation and techniques is described in Young and Gregory.[2]

0.	None	
1.	QO.1.1	Euler
2.	QO.1.2	Heun
3.	QO.1.3	Heun
4.	QO.1.4	Adam-Moulton (Adam-Bashforth)
5.	QO.2.1 & QO.2.2	Midpoint
6.	QO.2.3	
7.	QO.2.4	
8.	QO.3.2	
9.	QO.3.3	
10.	QO.3.4	
11.	QO.4.3 & QO.4.4	
12.	QO.1.1 with correction factor (Goel/Morris)	
13.	QO.1.2 with correction factor (Goel/Morris)	
14.	QO.1.3 with correction factor (Goel/Morris)	
15.	QO.1.4 with correction factor (Goel/Morris)	

Techniques 1-11 are explained by Young and Gregory.[2] Techniques 1-4 were modified slightly to adjust for the difference between the extrapolated positions and the new updated positions subject to some limits and number of frames between updates. These are techniques 12-15. The experiments confirmed that these modifications to the techniques worked very well during maneuvers. The Euler method was used for extrapolation of attitude, which also worked very well.

The parameters used in aircraft simulation were positions X, Y, Z; velocities $\dot{X}$, $\dot{Y}$, $\dot{Z}$; and accelerations $\ddot{X}$, $\ddot{Y}$, $\ddot{Z}$ along the three axes and, attitude parameters ϕ, θ, ψ and their rates P, Q, and R. The first twelve of these fifteen parameters are referenced in inertial coordinates except for P, Q, R which are in body coordinates. These body coordinates were transformed into inertial coordinates before using them in extrapolation algorithms. This transformation had potential singularities which will be discussed later. For Remote Vehicle Approximation (RVA) models, these fifteen parameters were not available for every frame. As a result, parameters such as position and attitude and possibly others were approximated from the most recent updated data received. The most simplistic and a very effective technique was the Euler iteration method shown below.

New Value = Old Value + Frametime x Rate (1)
of change per second

This technique was applied to approximate velocities from accelerations and positions from the newly approximated velocities, for both translational and attitude parameters. Various kinds of extrapolation techniques had been investigated. Some of which included multistep methods and higher order derivative extrapolations. These results are summarized in Table 1. The algorithms for these techniques are summarized in Figure 1 & 2. The Taylor series expansion of various order, multisteps, and higher order derivatives methods based on numerical quadrature were evaluated. None of these, which happen to be computationally expensive were any better than the simple Euler method. The result of the studies showed that the best technique was to extrapolate velocity, position, and attitude using the Euler method for fast moving vehicles. In some cases, particularly for slow moving vehicles, the second order Euler method may not be any better than the first order Euler as shown below.

	Updates/Sec	Position Error	Attitude Error
1st order Euler	0.434	2.48 ft	0.23 deg
2nd order Euler	0.459	2.76 ft	0.23 deg

Comparison of 1st & 2nd order in study flight

(Scenario 1)

For the experiments, the Euler method was modified slightly and a correction factor was added to adjust for the difference between extrapolated positions (X, Y, Z only) and the new updated buffer just received subject to some limits and number of frames since last update. This technique worked very well for maneuvers requiring an improved extrapolation algorithm (an improvement of 3.7% for 1st order extrapolation and 1.4% for 2nd order extrapolation, as shown in Table 2). From a computational point of view, it was very inexpensive. However the drawback is that one may have to experiment to find the best correction factor for a certain class of maneuvers.

For extrapolation of attitude, the simple Euler method is recommended. The algorithm to convert P, Q, R to $\dot{\phi}, \dot{\theta}, \dot{\psi}$ are:

$$\begin{aligned} \dot{\psi} &= (Q \sin\phi + R \cos\phi) / \cos\theta \qquad (2) \\ \dot{\theta} &= Q \cos\phi - R \sin\phi \\ \dot{\phi} &= P + \dot{\psi} \sin\theta \end{aligned}$$

The presence of the cos θ term in the denominator of the first equation would make it undefined if theta equals 90 degrees. In theory it could happen, but during these experiments, there were no noticeable glitches. (a 90 degree nose up or down attitude was attempted for relatively long periods).

Position error thresholds were implemented in two different ways.

$$ERR1 = Max \{ (X - X_{drk}), (Y - Y_{drk}), (Z - Z_{drk})\} \qquad (3)$$

$$ERR2 = ((X - X_{drk})^2 + (Y - Y_{drk})^2 + (Z - Z_{drk})^2)^{1/2} \quad (4)$$

Where $(X - X_{drk})$, etc. represent the delta between actual and dead reckoned models. Experimentation showed that technique 1 generates a lower number of updates. Because of ease of computation and better results, technique 1 was chosen and is recommended.

The difference between two orientations can be measured by an angle of rotation between two vehicle orientations. Pope describes this implementation.[5] This technique was implemented. The use of delta error for

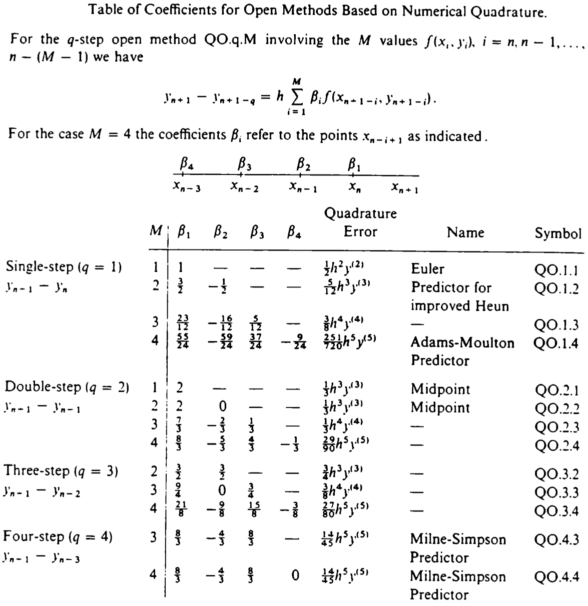


Table of Coefficients for Open Methods Based on Numerical Quadrature.

For the q-step open method QO.q.M involving the M values $f(x_i, y_i)$, $i = n, n-1, \ldots, n-(M-1)$ we have

$$y_{n+1} - y_{n+1-q} = h \sum_{i=1}^{M} \beta_i f(x_{n+1-i}, y_{n+1-i}).$$

For the case $M = 4$ the coefficients β_i refer to the points x_{n-i+1} as indicated.

$$\begin{array}{ccccc} \beta_4 & \beta_3 & \beta_2 & \beta_1 & \\ \hline x_{n-3} & x_{n-2} & x_{n-1} & x_n & x_{n+1} \end{array}$$

	M	β_1	β_2	β_3	β_4	Quadrature Error	Name	Symbol
Single-step ($q = 1$)	1	1	—	—	—	$\frac{1}{2}h^2y^{(2)}$	Euler	QO.1.1
$y_{n+1} - y_n$	2	$\frac{3}{2}$	$-\frac{1}{2}$	—	—	$\frac{5}{12}h^3y^{(3)}$	Predictor for improved Heun	QO.1.2
	3	$\frac{23}{12}$	$-\frac{16}{12}$	$\frac{5}{12}$	—	$\frac{3}{8}h^4y^{(4)}$	—	QO.1.3
	4	$\frac{55}{24}$	$-\frac{59}{24}$	$\frac{37}{24}$	$-\frac{9}{24}$	$\frac{251}{720}h^5y^{(5)}$	Adams-Moulton Predictor	QO.1.4
Double-step ($q = 2$)	1	2	—	—	—	$\frac{1}{3}h^3y^{(3)}$	Midpoint	QO.2.1
$y_{n+1} - y_{n-1}$	2	2	0	—	—	$\frac{1}{3}h^3y^{(3)}$	Midpoint	QO.2.2
	3	$\frac{7}{3}$	$-\frac{2}{3}$	$\frac{1}{3}$	—	$\frac{1}{3}h^4y^{(4)}$	—	QO.2.3
	4	$\frac{8}{3}$	$-\frac{5}{3}$	$\frac{4}{3}$	$-\frac{1}{3}$	$\frac{29}{90}h^5y^{(5)}$	—	QO.2.4
Three-step ($q = 3$)	2	$\frac{3}{2}$	$\frac{3}{2}$	—	—	$\frac{3}{4}h^3y^{(3)}$	—	QO.3.2
$y_{n+1} - y_{n-2}$	3	$\frac{9}{4}$	0	$\frac{3}{4}$	—	$\frac{3}{8}h^4y^{(4)}$	—	QO.3.3
	4	$\frac{21}{8}$	$-\frac{9}{8}$	$\frac{15}{8}$	$-\frac{3}{8}$	$\frac{27}{80}h^5y^{(5)}$	—	QO.3.4
Four-step ($q = 4$)	3	$\frac{8}{3}$	$-\frac{4}{3}$	$\frac{8}{3}$	—	$\frac{14}{45}h^5y^{(5)}$	Milne-Simpson Predictor	QO.4.3
$y_{n+1} - y_{n-3}$	4	$\frac{8}{3}$	$-\frac{4}{3}$	$\frac{8}{3}$	0	$\frac{14}{45}h^5y^{(5)}$	Milne-Simpson Predictor	QO.4.4

FIGURE 1. YOUNG & GREGORY - A SURVEY OF NUMERICAL MATHEMATICS VOL. 1

roll, pitch, and yaw angles was investigated with similar results but lower computational load.

Delays up to 20 frames were implemented. The same extrapolation techniques were used for both dead reckoning and delay compensation. Using the time stamp provided in the received Appearance PDU, the received position was extrapolated to the current position (or attitude) during one time frame and thereafter updated at frame rate. A switch was implemented to allow the option of turning the delay compensation ON/OFF. To simulate the various time delays, a circular buffer technique was implemented which allowed a different amount of delay from 0 to 1 second in increments of 50msec (our frame time) for different dead reckoning models. Each model had a choice of compensating or not compensating for the delay subject to flags in the software. The delay parameters were initially read from a data file and were changed on-line in real time to study the effects.

Five different smoothing techniques were evaluated in each run. Four tables were created to store the rate of adjustment at each step for the differences between the extrapolated position and the visual world coordinate sets. The number of steps used and the amount of compensation applied were subject to the adjustment needed and the associated table. A running total of the adjustment applied was kept to stop excessive adjustment if updates were too far apart. There was the flexibility of upscaling/downscaling of the adjustments using touch screen inputs. Jumps more than five feet along any axes and any jumps more than five degrees along roll axis were not allowed to avoid excessive jerkiness.

Big jumps were allowed if the difference was more than a pre-set envelop to avoid excessive drifting of world coordinate set from the actual flight path (a large envelop of 200 feet and 120 degrees was used). This phenomenon did not happen often except at initial

COMPARISON OF UPDATES FOR GOEL/MORRIS VS EULER						
SCN #	1st Order		Advntg.	2nd Order		Advntg.
	G/M	Euler		G/M	Euler	
1	0.468	0.429	-9.09	0.546	0.484	-12.91
2	0.758	0.870	12.84	0.575	0.559	-2.86
3	1.700	1.797	5.42	0.744	0.744	0.0
4	2.005	2.114	5.18	0.854	0.877	2.68
5	3.001	3.087	2.90	1.013	1.076	5.84
6	3.188	3.290	3.11	0.960	0.015	5.42
7	2.407	2.532	4.94	0.766	0.802	4.59
8	2.177	2.302	5.43	0.934	0.928	-0.64
9	3.588	3.660	1.98	1.830	1.866	1.94
10	2.067	2.157	4.20	1.029	1.062	3.10
11	3.601	3.770	2.63	2.464	2.513	1.97
12	2.702	2.790	3.18	2.214	2.197	-0.81
Avg.	2.311	2.400	3.71	1.161	1.177	1.38

These Data Represent the Updates/Sec Occurring During Each Scenario With and Without a Correction Factor. Comparisons are Either First or Second Order Extrapolation With a Difference Percentage Given for Each Scenario. The Bottom Line Represents an Overall Average. The Update Envelope Used Was 9.1 Feet and 3 Degrees.

TABLE 2

start. Various techniques (Hermit interpolation, using 100 frame projection for smoothing) were tried but the implemented technique was more dynamic and kept the dead reckoning model closer to the actual flight path. The position and attitude error results for twelve scenarios, using 2nd order extrapolation for delay compensation, and smoothing techniques are summarized in Table 3.

CONCLUSIONS

Most experience in Northrop's Flight Simulation Laboratory was with high speed aircraft simulations. With this background it is easy to have a skeptical point of view when considering the possibility of using high fidelity simulations over long haul networks. However, this study has shown that using second order Dead Reckoning and smoothing algorithms produces reasonable performance with low update rates at delays of up to 750 msec. These investigations point to several subjects of equal importance. First, in the majority of cases, 2nd order extrapolation proved superior both in positional accuracy and update rate (the exception being straight and level flight). Secondly, the use of a simple Euler extrapolation algorithm produced satisfactory results for dead reckoning and delay compensation. Third, the use of a smoothing algorithm becomes more important as transmission delays increase.

Both 1st and 2nd order extrapolation were tried and found an improvement of about 2 to 1 with 2nd order. A similar improvement factor was also present when average position errors were considered. In the case of straight and level flight, the 1st order extrapolation worked better than the 2nd order extrapolation. For delay compensation, a time tag is sent in each update message. This is used to update position and attitude parameters to the correct extrapolated values at the time a message is received. This produces a very close extrapolated position on both sides of the network between the time a message is received and the time a new one is transmitted.

Several different extrapolation algorithms were tested. These included using up to four previous steps to estimate the future path. In each case these additional steps would either have to be included in each Appearance PDU or calculated by the RVA. Both solutions were tested. The higher order extrapolation techniques (i.e., multi-step) will demand an increase in network bandwidth utilization to provide the required additional number of previous flight parameters. This will incur an increased computational load at each site. The preliminary experimentation indicates that the insignificant improvements provided by these techniques may not warrant their adoption. When a network update occurs, a jump to the new delay corrected position is apparent. With long delay times (500 ms to 750 ms) these jumps can be significant. Certainly, with any visual tracking a jump is unsatisfactory. Its effect on radar or missile tracking algorithms is not known at this time.

The effect of maneuvering upon update rate can be examined with the use of specialized graphs, as shown in Figure 2. The top part of the graph represents a specific maneuver while the lower portion shows network updates for first and second order extrapolation in updates per second. The top line, labeled with a v,

SECOND ORDER EXTRAPOLATION UPDATES AND ERRORS DUE TO DELAY COMPENSATIONS										
SCN #	UPDATES PER SEC	ERROR	0 MSEC		150 MSEC		500 MSEC		750 MSEC	
			SMOOTHING		SMOOTHING		SMOOTHING		SMOOTHING	
			OFF	ON	OFF	ON	OFF	ON	OFF	ON
1	0.459	POS	2.76'	4.57'	3.35'	5.40'	5.19'	10.08'	10.08'	13.10'
		ATT	0.23°	0.23°	0.31°	0.31°	0.61°	0.61°	0.91°	0.91°
2	0.533	POS	2.98'	5.16'	3.79'	6.27'	7.75'	10.74'	12.60'	15.86'
		ATT	0.18°	0.18°	0.21°	0.21°	0.29°	0.29°	0.35°	0.35°
3	0.744	POS	2.82'	6.14	3.93'	8.29'	9.25'	17.15'	16.61'	27.66'
		ATT	0.26°	0.26°	0.42°	0.42°	1.00°	1.01°	1.55°	1.56°
4	0.811	POS	2.70'	7.04'	4.04'	9.99'	10.43'	21.99'	18.44'	33.86'
		ATT	0.51°	0.51°	0.73°	0.73°	1.45°	1.45°	2.11°	2.11°
5	1.016	POS	2.43'	8.45'	4.10'	13.07'	12.04'	30.75'	22.83'	50.93'
		ATT	0.23°	0.23°	0.38°	0.38°	0.90°	0.90°	1.38°	1.38°
6	0.943	POS	2.55'	7.80'	4.08'	11.70'	10.38'	25.60'	20.76'	43.45'
		ATT	0.24°	0.25°	0.41°	0.42°	1.02°	1.05°	1.52°	1.54°
7	0.770	POS	2.90'	7.75'	4.33'	10.94'	9.66'	21.73'	18.30'	35.43'
		ATT	0.38°	0.38°	0.54°	0.54°	1.07°	1.07°	1.57°	1.57°
8	0.844	POS	2.46'	6.47'	3.74'	9.40'	9.90'	20.95'	17.96'	34.23'
		ATT	0.28°	0.28°	0.43°	0.43°	0.90°	0.90°	1.33°	1.33°
9	1.836	POS	2.24'	8.51'	4.46'	15.19'	16.08'	42.90'	31.41'	74.92
		ATT	0.49°	0.49°	1.44°	1.45°	6.43°	6.44°	12.10°	12.12°
10	1.062	POS	2.69'	6.97'	4.32'	10.22'	11.36'	21.68'	20.35'	34.63'
		ATT	0.61°	0.61°	0.99°	1.00°	2.62°	2.63°	4.39°	4.39°
11	2.513	POS	1.99'	7.50'	4.86'	14.18'	20.19'	44.01'	39.51'	79.95'
		ATT	0.58°	0.58°	2.47°	2.47°	13.29°	13.30°	24.75°	24.76°
12	2.258	POS	0.98'	3.68'	1.87'	6.45'	6.36'	18.00'	12.47'	31.16'
		ATT	1.03°	1.03°	2.52°	2.53°	8.78°	8.78°	14.32°	14.32°
Avg.	1.149	POS	2.46'	6.67'	3.90'	10.09'	10.72'	23.55'	20.11'	39.60'
		ATT	0.42°	0.42°	0.90°	0.91°	3.20°	3.20°	5.52°	5.53°

Average Position in Feet and Attitude Error in Degrees, Between High Resolution Flight Path and Its RVA After Delay Compensation With Smoothing Off/On. The Update Envelope Used Was 9.1 Feet and 3 Degrees.

TABLE 3

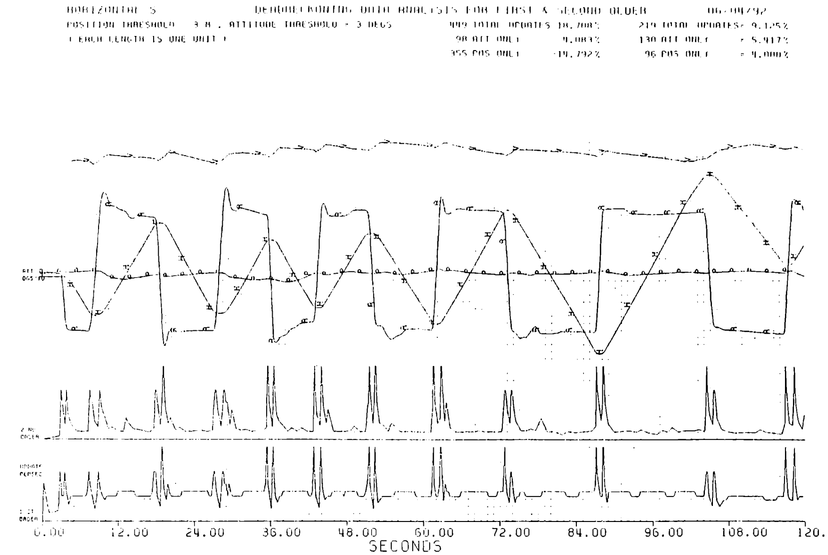


FIGURE 2. DEAD RECKONING DATA ANALYSIS FOR FIRST AND SECOND ORDER EXTRAPOLATION

indicates a total velocity and shows any speed changes. The next three are roll, pitch, and heading (R, P, and H). These give a history of attitude during the entire trial period (two minutes). The slopes of the lines represent rates. It can be seen that roll produces the highest rate and is therefore the high frequency parameter that has the most profound effect on update rate. The update rates are represented by the two lower graphs with first order being the bottom. Each vertical line above the two baselines is equivalent to one update per second. For instance, at the 12 second mark, first order is 4/sec while second order is slightly higher than 1/sec. For both orders, spikes to about 10/sec occur at high roll rate points. This follows from both sets using first order for attitude and either first or second order for position. The primary result is that over the vast majority of the test, first order averaged about 4/sec while second order averaged about 1/sec. This study has shown that Distributed Interactive Simulation can be accomplished with high speed aircraft simulations.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Leslie Klein and Mrs. Farida Patel of Northrop Corporation for their invaluable assistance in the preparation of this paper. We would also like to thank Lt. Col. David Greschke of 405 TTS / Tac Aces at Luke AFB, AZ who provided the aerial combat engagements from the Simulator for Air to Air Combat.

REFERENCES

(1) Morris K., and S. Goel; Interim Report For Long Haul Data Transmission System (LHDTS) Results; NOR 91-28; 1991.

(2) Young D., and R. Gregory; A Survey Of Numerical Mathematics; Vol I; Addison - Wesley Publishing Company; 1972.

(3) Thelander J.; Aircraft Motion Analysis; Technical Documentary Report; FDL-TDR-64-70.

(4) McDonald B.; Development Of Improved Dead Reckoning Algorithms For Distributed Interactive Simulation; White Paper presented to DMSO; July 1991.

(5) Pope A.; The SIMNET Network and Protocols; BBN Systems and Technologies Corporation; July 1989, Report No. 7012.

AIAA-92-4172-CP

THE VISTA/F-16 PROGRAMMABLE FEEL SYSTEM

R.J. Siracuse† and K.S. Govindaraj††
Arvin/Calspan Corporation, Buffalo, New York

ABSTRACT

This paper represents the culmination of a three year design and development program to fabricate a programmable force feel system to be installed in the USAF Variable Stability In-Flight Simulator Test Aircraft (VISTA). The VISTA, currently under development by General Dynamics and Arvin/Calspan, is a modified F-16 that can simulate the control, feel and dynamic characteristics of other aircraft. It is intended to replace the NT-33A, the USAF current fighter task in-flight simulator. A wide range of flying characteristics is provided by the VSS control system implemented on three on-board HAWK/32 computers. Simulation of the variable feel characteristics is provided by the programmable Force Feel System (FFS). The FFS consists of a dedicated microcomputer, Titan SECS 80/ATR, and a two axis servo controlled centerstick installed in the simulation (front) cockpit of a two place F-16. The standard F-16 sidestick was left intact in the simulation cockpit to provide both centerstick and sidestick capability during VSS operation.

The programmable FFS was integrated with the VISTA, and ground simulation tests were completed in the General Dynamics simulation facility, and subsequently in the VISTA F-16. In-flight testing is scheduled when the VISTA flight tests resume.

1. INTRODUCTION

The FFS consists of a two-axis electrohydraulic servo assembly and a digital model-following system implemented in a dedicated Titan SECS 80/ATR microcomputer. A HAWK/32 computer is used as an interface between the pilot and the microcomputer and for data storage of the variable feel configurations.

The analog electrohydraulic position servos have fixed second-order dynamic characteristics determined by rate and position feedback. The gains were selected to provide fixed static and dynamic characteristics (frequency and damping). The desired range of feel characteristics is achieved by the model programmed in the Titan microcomputer. This system is unlike the systems previously developed by Calspan, where the variable characteristics were generated by varying the feedback and command path gains.

The digital system generates servo commands that provide the variable static and dynamic characteristics. A second-order model is implemented in the computer together with the desired nonlinear characteristics. The desired stick motions result from pilot force inputs to the model. The control algorithm uses the model responses and the fixed analog servo damping and frequency to compute the servo command. The model includes the following adjustable characteristics:

- Frequency, damping ratio and force gradient
- Nonlinear gradient
- Hard position limit
- Preload
- Friction
- Freeplay
- Downspring and bobweight effects

A Configuration Control System (CCS) that resides in the HAWK computer is used to load pre-selected configurations and perform on-line modification of feel system parameters.

Figure 1 is a simplified block diagram of the VISTA two-axis centerstick feel system (pitch and roll). The pilot force in each axis is measured and transmitted to the digital model through an A/D converter, where it is processed by the control algorithm that computes the digital servo command. All of the desired servo characteristics are included in this computation. The resulting servo command is transmitted to the servo through a D/A converter. The lead/lag network, shown in Figure 1, computes an incremental command that provides servo stability at low force gradients (high loop gain). The total servo command is computed as a sum of the digital

†Principal Electrical Engineer
††Principal Systems Engineer
Member AIAA

servo command and the incremental command due to the lead/lag network. The commanded servo response provides the required static and dynamic characteristics to the pilot.The digitally controlled variable feel servo offers several advantages over previous analog mechanizations. These are briefly outlined as follows:

- Different variable feel features can be implemented in software without hardware changes. An example is transposition of the nonlinearities in the block diagram.
- Unlike previous systems in which the servo loop gain was used to control the the control system gradient, the analog position and rate feedback gains are set to fixed values. This allows the servo performance to be optimized, resulting in identifiable servo dynamics that are required for the model-following algorithm. It also reduces the deleterious effects of servo valve nonlinearities at low position feedback gains required for light stick forces.
- The electronics are simplified. The analog system developed by Calspan for use on the Variable Stability Learjet require 12 multiplying D/A converters per axis for control of the servo characteristics and non-linear parameters. The hybrid system developed for the VISTA reduced this requirement to two D/A and converters per axis.
- The variable feel nonlinear characteristics can be precisely modeled. Rounded corners on the nonlinear functions inherent in analog mechanization are eliminated.

The remainder of this paper is organized as follows: Section 2 describes the analog servo configuration; Section 3 describes the digital system; Section 4 describes the implementation of the complete system; and Concluding Remarks are given in Section 5.

2. SERVO CONFIGURATION

In the feel systems previously developed by Calspan, the electro-hydraulic actuators were configured as second-order systems with variable feedback gains. The servo position gain was used to control both the servo frequency and gradient; the rate feedback controlled the damping ratio. For the VISTA, the dynamics are generated by a second-order model programmed in the digital computer; the hydraulic servo characteristics are fixed. It is important that the servo is accurately configured and that higher-order effects are minimized by proper component sizing and gain

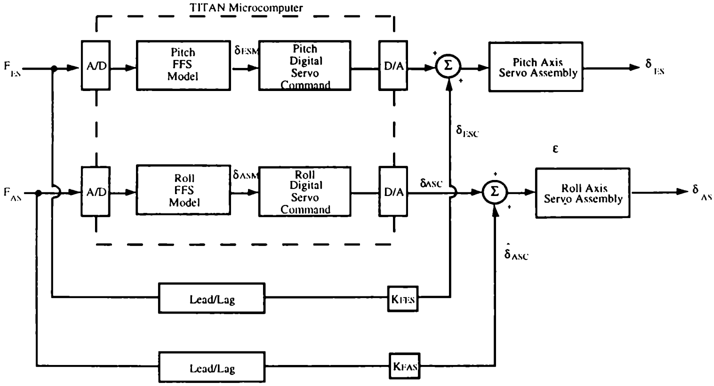
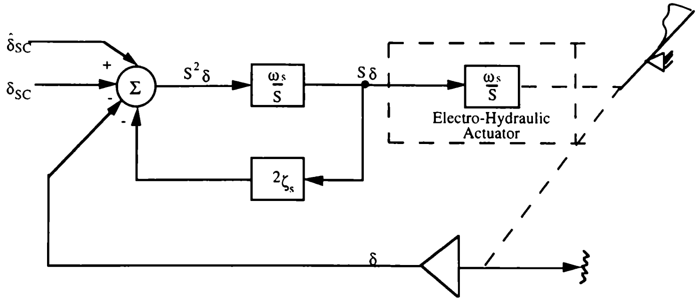


Figure 1 FFS SIMPLIFIED BLOCK DIAGRAM



ω_s	=	Undamped frequency	(rad/sec)
ζ_s	=	Damping ratio	-
δ_{SC}	=	Servo position command	(in)
δ	=	Servo position	(in)

Figure 2 FFS POSITION SERVO BLOCK DIAGRAM

distribution in the hydraulic servo control loop. The FFS position servo block diagram (for both pitch and roll) is shown in Figure 2. The analog position and rate feedbacks were set to achieve the following closed-loop dynamic characteristics for both axes:

$$\omega_s = 50 \text{ rad/sec}$$
$$\zeta_s = 0.9 \qquad (1)$$

The FFS was integrated with the General Dynamics' hotbench facility for performance tests. In the roll axis, the performance tests were successfully completed with the fixed servo configuration. In the pitch axis, structural coupling through the cockpit floor caused stability problems when the model bandwidth was different from the servo by a factor of two or more. This made it necessary to change the servo frequency in harmony with the model. Fixed analog servo characteristics were maintained by implementing an outer loop digital position feedback path. The incremental position feedback was controlled by the Titan and combined with the output of the model-following algorithm to modify the servo bandwidth.

The servo position feedback gain was scheduled as a function of model frequency as follows:

$35 < \omega_m \leq 50$	$\omega_s = 50$
$25 < \omega_m \leq 35$	$\omega_s = 35$
$\omega_m \leq 25$	$\omega_s = 25$

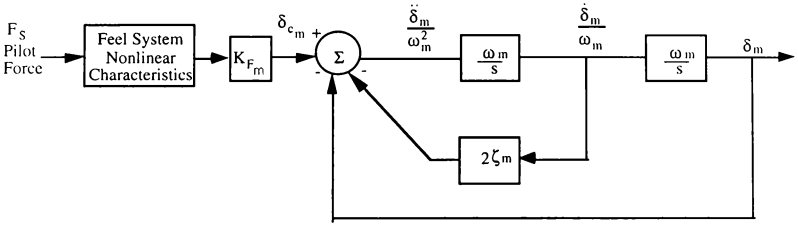
3. DIGITAL FEEL SYSTEM

The variable feel system characteristics for both pitch and roll axes are implemented in the Titan microcomputer. The desired dynamics (for both axes) are modeled by a second-order transfer function with independent control over gradient, natural frequency and damping ratio. The model transfer function and block diagram are shown in Figure 3. The force gain, K_{F_m}, is computed as the inverse of the specified gradient. The measured pilot force produces the model position command signal, δ_{mc}.

Tustin's integration technique is used in the dynamic equations, converting the integrator into the following discrete form:

$$s = \frac{2}{T}\frac{z-1}{z+1} \tag{2}$$

where T is the sample or computation time. Using the Tustin transformation, the integrators



$$\delta_m = \frac{K_{F_m}}{\left(\frac{s}{\omega_m}\right)^2 + \frac{2\zeta m}{\omega_m}s + 1}$$

Where:

δ_m ~ Model position	(in.)
F_s ~ Pilot stick force	(lb)
K_{F_m} ~ Force gain	(in./lb)
ζ_m ~ Damping ratio	-
ω_m ~ Model frequency	(rad/sec)
$1/K_{F_m}$ ~ gradient	(lbs/in.)
δ_{c_m} Model position command	(in.)

Figure 3 MODEL BLOCK DIAGRAM

shown in Figure 3 are represented by the following expression:

$$G(z) = \frac{\omega_m T}{2} \frac{z+1}{z-1} \qquad (3)$$

The model acceleration and rate are integrated using the difference equations derived from (3):

$$\left(\frac{\ddot{\delta}_m}{\omega_m^2}\right)_i = \delta_{c_{m_i}} - \delta_{m_{i-1}} - 2\zeta_m \left(\frac{\dot{\delta}}{\omega_m}\right)_{i-1}$$

where:

$$\delta_{c_{m_i}} = K_{F_m} F_{s_i}$$

$$(4)$$

$$\left(\frac{\dot{\delta}_m}{\omega_m}\right)_i = \left(\frac{\dot{\delta}_m}{\omega_m}\right)_{i-1} + \frac{\omega_m T}{2}\left[\left(\frac{\ddot{\delta}_m}{\omega_m^2}\right)_i + \left(\frac{\ddot{\delta}_m}{\omega_m^2}\right)_{i-1}\right]$$

$$\delta_{m_i} = \delta_{m_{i-1}} + \frac{\omega_m T}{2}\left[\left(\frac{\dot{\delta}_m}{\omega_m}\right)_i + \left(\frac{\dot{\delta}_m}{\omega_m}\right)_{i-1}\right]$$

where the subscript i denotes the present value, and (i-1) denotes the past value.

The feel system and nonlinear characteristics such as the preload and friction are included in the force command path shown in Figure 3. The digital servo command is computed using the feel system servo parameters as follows:

$$\delta_{C_D} = \left[\left(\frac{s}{\omega_s}\right)^2 + \frac{2\zeta_s}{\omega_s} s + 1\right]\delta_m \qquad (5)$$

This control law, combined with the servo transfer function in Figure 3, results in a unity transfer function between the servo position and model position.

The discrete representation of equation (5) is given by

$$\delta_{C_{D_i}} = \left(\frac{\ddot{\delta}_m}{\omega_s^2}\right)_i + 2\zeta_s\left(\frac{\dot{\delta}_m}{\omega_s}\right)_i + \delta_{m_i} \qquad (6)$$

4. FEEL SYSTEM IMPLEMENTATION

The programmable feel system consists of an analog servo with fixed dynamic characteristics, a digital model that provides the desired variable feel characteristics, and a model-following control algorithm. This section discusses the integration of these subsystems.

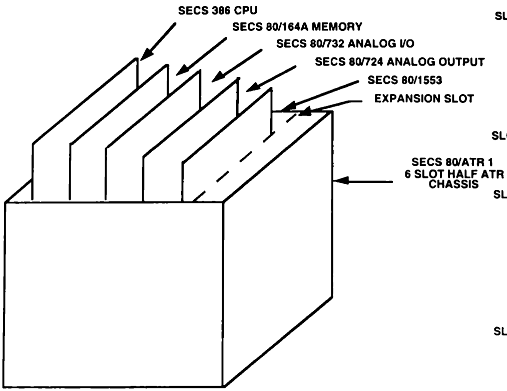
The digital feel system concept was implemented, developed, and tested using hardware in the Total In-Flight Simulator (TIFS) operated by Calspan for the U.S. Air Force. The objectives of the TIFS tests were to determine the feasibility of the digital feel system and determine the software, hardware, and update requirements. The tests demonstrated the feasibility of the digital feel system and provided the system goals and requirements for VISTA implementation. The system implementation is summarized as follows:

- Microcomputer
- I/O requirements
- Servo stability compensation
- Servo nonlinear characteristics
- Sensors
- Microcomputer interface

A summary of these requirements is given in this section.

Microcomputer

The TIFS proof of concept tests established the computer requirements. The criteria included: (1) computational capability (2) sufficient I/O – a minimum of 6 D/A converters and 16 A/D converters were required for the 2 axis centerstick (3) speed – a design goal of 2 milliseconds was specified as the update rate for the 2 axis centerstick. The computer also had to satisfy MIL standards for temperature and vibration. The HAWK/32 was ruled out because the update rate could not be satisfied. As a result a dedicated microcomputer was selected. The Titan SECS 80/ATR was selected for several reasons. It was an "off the shelf unit", satisfied the speed requirement and had expansion capability. The Titan, as configured for the VISTA FFS, is shown in Figure 4.



TITAN
MODEL, SECS 386/20

MANUFACTURED IN ACCORDANCE WITH:

- MIL-E-5400
- MIL-E-16400
- MIL-E-4158
- DOD-E-8983

SLOT 1 SECS 386 CPU

- 32 BIT COMPUTER WITH 16 MHZ 80386/80387 MICROPROCESSORS
- 128K BYTES EPROM

SLOT 2 SECS 80/164A MEMORY

- UP TO 2M BYTES STATIC RAM

SLOT 3 SECS 80/732 ANALOG I/O

- 12 BIT A/D AND D/A CONVERSION
- 32 SINGLE ENDED OR 16 DIFF A/D INPUTS
- 2 D/A OUTPUTS
- 4 PROGRAMMABLE GAINS

SLOT 4 SECS 80/724 ANALOG OUTPUT

- 4 12 BIT D/A OUTPUTS

SLOT 5 SECS 80/1553

- DUAL REDUNDANT 1553 MULTIPLEX BUS

Figure 4 FEEL SYSTEM COMPUTER TITAN SECS 386 MICROCOMPUTER SYSTEM

Servo Stability Compensation

Based on variable feel system experience, it is known that lead compensation is required in the force command path to preserve servo stability at light force gradients; i.e., high force gains in the forward path. A variation of this compensation was devised for the digital feel system in which a parallel analog path was used to generate the required lead compensation. A simplified block diagram of the digital feel system with the servo stability compensation is shown in Figure 5. The net servo command is given as:

$$\delta_C = \hat{\delta}_{SC} + \delta_{SC} \tag{7}$$

Where:

$$\delta_{SC} = \delta_{CD} - \delta_D$$

$$\hat{\delta}_{SC} = K_{FS}\, F_p$$

$$\delta_D = K_{FS}\, F_p \tag{8}$$

The servo force to position gain, K_{F_S}, is set by the digital control algorithm. This gain is scheduled as a function of the model frequency and is proportional to the inverse of the gradient. This parameter also controls the gain of $\hat{\delta}_{SC}$, the analog compensation variable.

Servo Nonlinear Characteristics

The nonlinear feel characteristic wave shapes are shown in Figure 6. These characteristics are programmable and used in conjunction with the

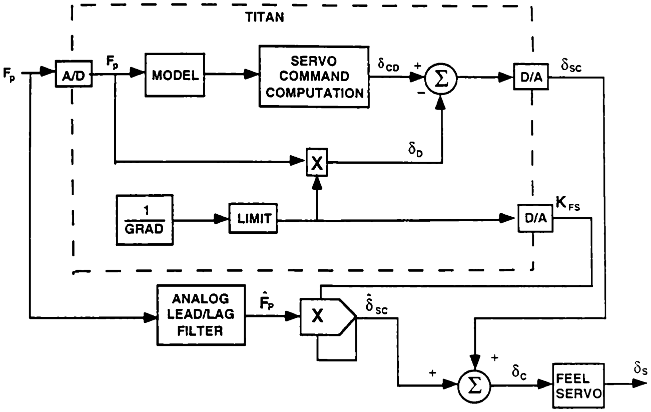


Figure 5 FEEL SYSTEM WITH SERVO STABILITY COMPENSATION

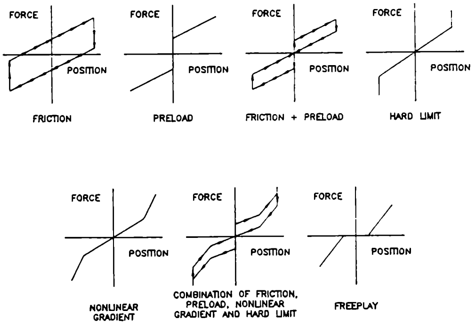


Figure 6 FEEL SYSTEM NONLINEARITY CHARACTERISTICS

second-order model implemented in the Titan microcomputer. The nonlinear characteristics can be varied independently and may be selected individually or used in combination. A block diagram that includes the second-order model, all the nonlinearities, and the bobweight and downspring effects is shown in Figure 7. The preload, friction and nonlinear gradient characteristics operate on the force signal to shape the servo command. The hard limit and freeplay are programmed as a function of the model position signal. Bobweight and downspring effects are generated as a function of pitch acceleration, normal acceleration and dynamic pressure. The maximum and minimum values of the programmable servo characteristic parameters are summarized in Table 1. The stick trim is under software control and can be implemented as a force or position mode. When configured in the force trim mode, the trim signal is combined with the pilot force input and is affected by all of the feel system nonlinearities. When configured in the position trim mode, the trim signal bypasses all nonlinearities and is input as a stick position command.

Sensors

Each axis has dual force and position sensors to provide fail safe operation. Comparitors in the Titan and HAWK's automatically disengage the VSS if a mismatch is detected between the dual parameters. When the VSS is disengaged, the feel system is disabled and control of the aircraft reverts to the safety pilot in the rear cockpit.

I/O Requirements

The Titan interface presently consists of the following I/O ports:

- 32 single input A/D converters
- 6 D/A converters
- 8 discrete inputs
- 8 discrete outputs

Thirteen A/D converters are used in the VISTA to transmit the dual force and position signals, servo rate, acceleration, and the trim signal to the Titan. The remaining A/D converters will be utilized for future expansion. Two of the Titan D/A converters are used to command the pitch and roll servos. The servo compensation gains are controlled by two other D/A converters. The remaining two D/A converters are presently used to output pitch and roll model positions for checkout purposes. The discrete inputs and outputs are used to control various software and hardware functions.

Titan Interface

A 1553 bus interface provides the communication between the Titan and the HAWK computers. The feel system configurations are stored in the HAWK computer. A configuration control system residing in the HAWK allows the pilot to select any pre-stored configuration. In addition, the pilot can request a change to any of the feel system parameters or configurations during the flight evaluations. The pilot-requested information is communicated to the Titan through the 1553 bus interface. The Titan accepts the pilot-initiated request for change and communicates back to the HAWK through the 1553 bus interface following the completion of the requested changes.

A detailed block diagram of the VISTA force feel system is shown in Figure 8. A brief review of the system is given as follows:

- Pilot Force Input: The pitch and roll force inputs, F_{ES} and F_{AS} respectively, are transmitted to the microcomputer for processing. The force inputs are also processed by the analog lead/lag stability compensation filters. Capability exists to replace the pilot force inputs with test force inputs.
- Stability compensation: The analog lead/lag filter that provides stability compensation at light force gradients is shown for the pitch axis. This analog path generates the pitch and roll compensation signals that are summed with the model-following commands generated by the microcomputer. The compensation gains are controlled by the computer.
- Feel system servo: The feel system servo shown includes the servo actuator, unity position feedback and rate feedback. Only the pitch axis servo system is shown. The servo pitch and roll position, rate and acceleration signals are transmitted into the computer for safety trip computations.

- Digital computer: The desired feel system characteristics for both the pitch and the roll axes are implemented in the microcomputer. These include the second-order dynamic model, all the nonlinear characteristics, stick trim computations, and the servo command computations. The pitch and roll servo commands, δ_{ESC} and δ_{ASC}, and the stability compensation gains, K_{FES} and K_{FAS}, are transmitted through the D/A converters.
- VMUX 1553 Interface: The 1553 VMUX bus provides the interface between the Hawk computer and the microcomputer. Parameter and configuration change requests initiated by the pilot are communicated to the microcomputer by the Hawk computer.

CONCLUSIONS

The variable force feel system developed for the VISTA is a unique application of the model-following control concept. It was successfully demonstrated during the proof of concept tests performed in the TIFS aircraft. The VISTA feel system was integrated with the General Dynamics' hotbench facility and performance tests were successfully completed. It was then installed in the aircraft and the ground performance test was successfully completed. Flight testing will be initiated when VISTA flight tests resume.

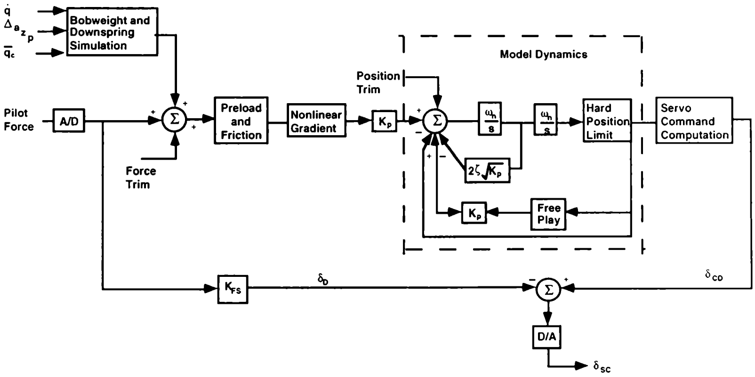


Figure 7 VARIABLE FEEL SYSTEM MODEL AND COMMAND COMPUTATION BLOCK DIAGRAM

Table 1. FEEL SYSTEM PARAMETER RANGE

UNITS	PITCH PARAMETERS	PITCH		ROLL	
		MAX VALUE	MIN VALUE	MAX VALUE	MIN VALUE
lb/in.	Gradient	* 100	0.1	* 100	0.1
rad/sec	Frequency	50	0.0	50	0.0
_	Damping Ratio	2	0.2	2	0.2
lb	Breakout Force	10	0.0	5.0	0.0
lb	Hysteresis	10	0.0	5.0	0.0
	Nonlinear Gradient				
lb	Breakpoint 1	50	> 0	50	> 0
–	Slope 1 Change	5	0.25	5	.25
lb	Breakpoint 2	100	>BP1	100	>BP1
–	Slope 2 Change	5	0.25	5	.25
in.	Stick Bias	4	-4	4	-4
in.	Foreward Limit	5	0	4	0
in.	Aft Limit	5	0	4	0
in.	Pos Trim Limit	±4	0	4	-4
in./sec	Pos Trim Rate	0.5	0	0.5	0
in.	Free Play	0.5	0	0.5	0
lb/sec	Force Trim Rate	10	0	10	0
lb	Force Trim Limit	80	0	40	0
	Bobweight,				
lb/g	N_z Gain	20.0	-20.0		
lb/deg/sec^2	$\dot{q}$ Gain	0.2	-0.2		
	Downspring				
lb/ft/sec	Velocity Gain	0.2	-0.2		

STIFF STICK (infinite gradient) is obtained by setting the gradient at a value larger than 100 lbs/in. The software is programmed to recognize this as infinite stiffness.

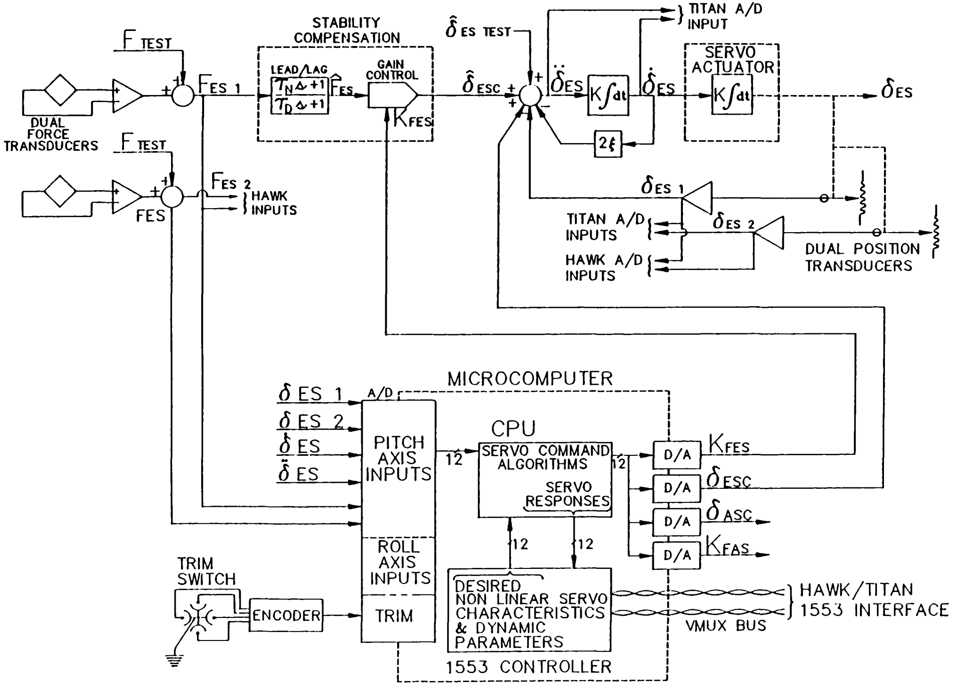


Figure 8 FFS BLOCK DIAGRAM (PITCH AXIS, TYPICAL)

AIAA-92-4177-CP

MAN-VEHICLE SYSTEMS RESEARCH FACILITY: DESIGN AND OPERATING CHARACTERISTICS

ROBERT J. SHINER *
BARRY T. SULLIVAN **
NASA-Ames Research Center
Moffett Field, California

ABSTRACT

This paper describes the full-mission flight simulation facility at the NASA Ames Research Center. The Man-Vehicle Systems Research Facility (MVSRF) supports aeronautical human factors research and consists of a building, two full-mission flight simulators and an air traffic control simulator. The facility is used for a broad range of human factors research in both conventional and advanced aviation systems. The objectives of the research are to improve our understanding of the causes and effects of human errors in aviation operations, and to limit their occurrence. The facility is used to:

1. Develop fundamental analytical expressions of the functional performance characteristics of aircraft flight crews;
2. Formulate principles and design criteria for aviation environments of the future;
3. Evaluate the integration of new subsystems in contemporary flight and air traffic control scenarios;
4. Develop new training and simulation technologies that are required by the continued technical evolution of flight systems and of the operational environment.

A cut-away view of the MVSRF is illustrated in Figure 1.

INTRODUCTION

Flight simulators have been increasingly indispensable tools for doing research in aeronautics and astronautics. Before flight test, the modern simulator is often used to provide verification of the proof of concept for new avionics, controls, or other systems designed to enhance the efficiency or performance of advanced aircraft and spacecraft. At the same time, simulators are replacing the use of aircraft for the training of flight crews, in some cases for reasons of economy, but also because certain kinds of training can be done in simulators which are impractical and dangerous in the actual aircraft. A central issue in simulation, whether it be used for research or training, is that of fidelity. For training applications, the requirements for fidelity are straightforward; a high degree of fidelity is only useful if it provides an effective training environment. There is no quantitative requirement for a given degree of fidelity, only for that which will produce the most rapid and long lasting training benefit. For simulators used in research, the requirements are different. The objective here is to generalize the results of studies done in the simulator to the actual aircraft and flight situation. Here the requirements for fidelity are more stringent because, by nature, research is used to explore the unknown; each study is to some degree different. Contemporary aeronautical human factors research may provide the most stringent requirements yet for certain kinds of simulation fidelity. Here the emphasis is on determining how flight crews behave in normal and abnormal situations: how they make decisions, manage workload, communicate among themselves and with ground personnel, and how they solve problems and process information. Experience at the MVSRF has pointed clearly to the need for a high degree of operational fidelity if the goal is for flight crews to behave in simulators like they do in the actual flying situation. A high degree of operational fidelity can be achieved through full-mission simulation of a functionally complete aircraft as well as the environment in which it operates. When this is done, the flight crew can perform the full range of behaviors in realistically complex flight scenarios. Experience has shown that crews do indeed perform in a remarkably similar fashion to those in actual flight, including making the same kind and number of errors during the course of a flight. The aeronautical human factors research program at Ames is broadly structured to address the problems with today's aviation system as well as the issues raised by the introduction of new technologies in the future. To provide capabilities in both areas, two flight simulators are used: one is representative of current technology and the second is representative of technology of the future.

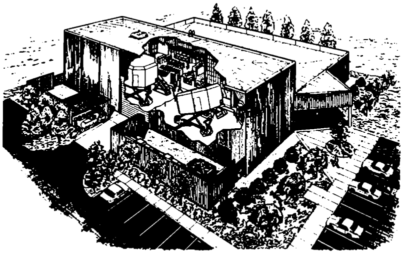


Figure 1 Cut-away view of the MVSRF

DESCRIPTION OF SIMULATORS

The facility is capable of supporting full-mission simulations representative of both current and future aviation operations. The Boeing 727 simulator, representing the current fleet, is an exact replica of the cockpit of a widely used domestic turbofan transport. Rigorous configuration control is maintained to ensure that aircrew behavior in simulated flights is representative of actual flight operations.

In contrast, the Advanced Concepts Flight Simulator (ACFS), configured with multiple electronic displays, advanced crew-aircraft interfaces and flight control devices, is designed to permit virtually unlimited flexibility in information presentation

* Chief, Full-Mission Simulation Branch
** Facility Manager for Advanced Concepts Flight Simulator, Member AIAA

and command and control by the aircrew. Such flexibility permits the simulation of operations representative of those found in advanced aircraft and air traffic control systems.

Both aircraft simulators can operate independently or in conjunction with the facility's Air Traffic Control Simulator. The ATC simulator can be configured to represent either today's aviation system or various possible systems of the future. The ATC simulator can be linked to run with either simulator, simultaneously with both simulators, or, it may be used in stand-alone operations.

BOEING 727-200 SIMULATOR

A key component of the facility is a Boeing 727-200 flight simulator. The simulator provides all modes of operation, including preflight, pushback, engine start, taxi, take-off, climb, cruise, descent, approach for landing, flare, touchdown, and park. All normal and many abnormal procedures can be simulated. Changes in aircraft attitude, thrust, drag, altitude, temperature, gross weight, center of gravity and configuration are accurately modeled. Ground effects are also modeled, including tire and brake effects with various runway conditions: dry, wet, icy, patchy wet, and patchy ice.

The simulator crew compartment is a full-scale replica of a current commercial airline cockpit; all instruments, controls, and switches operate as they do in the actual aircraft. All functional systems of the aircraft are simulated in accordance with aircraft data. A picture of the 727-200 simulator is shown in Figure 2.

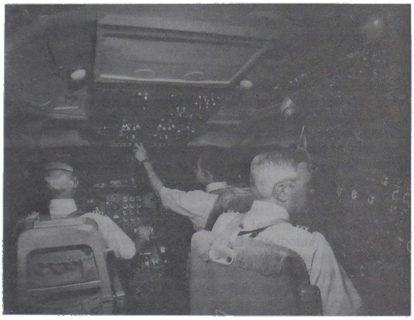


Figure 2 A view of the Boeing 727-200 Simulator

Sound cues are simulated using a controlled sound system. Sounds simulated include those of engine noises, aerodynamic noises, landing gear actuation, and runway effects. Auxiliary device sounds such as horns, bells and chimes are produced with actual aircraft hardware. Visual cues are provided by a Link Miles Image II computer generated image system, which provides the out-the-window visual scenes for the pilots. A six-degree-of-freedom synergistic system provides motion cues to the pilot to simulate normal flight cues as well as disturbances caused by bad weather. A hydraulic control-loading system simulates the characteristics of the aircraft's primary flight controls--wheels, columns and rudder pedals. Changes in the amount of movement and force on the controls are a function of aircraft acceleration, velocity, configuration, center of gravity and the type of control system specific to the aircraft. To assure the credibility of research results, the simulator is maintained to FAA Level C (Phase II) in accordance with FAR part 121, Appendix H and Advisory Circular AC120-40B.

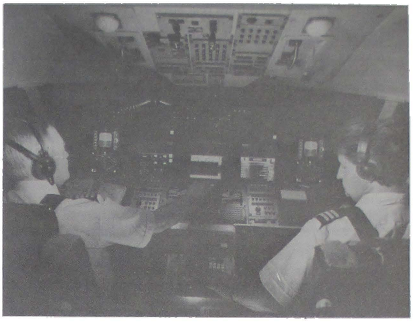


Figure 3 View of the ACFS

ADVANCED CONCEPTS FLIGHT SIMULATOR

The Advanced Concepts Flight Simulator (ACFS) is representative of a generic commercial transport aircraft based on projected user needs beyond the year 2000. This concept led to the design of the aircraft with the following characteristics:

- Maximum gross weight - 225,000 lbs.
- Payload - 60,000 pounds; capacity - 200 passengers
- Twin engine - 41,000 pounds rated thrust per engine
- Speed - 0.78 Mach; range - 2500 miles
- Flight crew - two persons
- All-electric airplane (no hydraulics except for nose wheel steering)
- Fly-by-wire; active flight controls
- Relaxed static margin; load alleviation
- T-tail, low wing, supercritical airfoil
- Composites for primary and secondary structures

The baseline flight station is designed for operation by a two-pilot crew. It is configured so that all controls pertinent to operation of the aircraft are accessible for operation by both pilots. A picture of the ACFS is shown in Figure 3. The simulator employs many features that exist in the newest aircraft flying today as well as advanced features that will be flying in future aircraft in the years to come.

Like some of the newer aircraft being built today the ACFS is representative of a "glass cockpit" arrangement. It is equipped with five cathode ray tube (CRT) monitors which electronically depict primary and secondary flight display information. It is also equipped with sidearm controllers for pitch and roll control, an integrated flight management system coupled to the aircraft's autopilot and autothrottle systems, and an electronic engine indication and crew alerting system (EICAS) which provides aircraft cautions, warnings and advisories to the flight crew. A unique feature about the ACFS flight displays is that they are reprogrammable and allow rapid prototyping of new types of aircraft displays, schematics or new types of aircraft systems depending on researcher requirements. In addition the flight management system is also programmable providing the capability to modify or upgrade the system to include advanced features such as 4-D time-based navigation or data-linking to air traffic control. Other advanced features that have been incorporated into the ACFS baseline include the following: touch sensitive screens on the secondary flight displays which enable the flight crews to go from one aircraft system or schematic to another, simply through the touch of a button on the display screen or menu; an advanced terminal weather information display which depicts microburst and wind shear activity uplinked from a ground based Doppler weather radar system; electronic checklists tied into phase of flight and reminder logic to alert the flight crew to recheck missed items; an integrated taxiway display which depicts airport traffic and ground clearances uplinked from air traffic control; a three dimensional spatialized audio system which provides localized communications, malfunctions and traffic warnings and/or alerts to the flight crews headsets.

727/ACFS VISUAL SYSTEM CAPABILITIES

Both the 727 and ACFS simulators utilize the same visual system. The Image II visual system is a compact flight simulator attachment which presents computer-generated color scenes representing the outside world. These scenes normally depict specific airports and their surroundings, as viewed at dusk or night from the cockpit. Enroute visual scenes are also simulated.

Night scenes include light points, horizon glow, runway markings and textures in the area illuminated by the simulated aircraft's landing lights. In dusk conditions, ground surface and building textures are shown in addition to the light points. At all times, occulting of lights, surfaces, and horizon glow by intervening surfaces is simulated.

Thirty-two occulting surfaces and levels are provided. The system can create strings of random lights in addition to curved strings.

Airport scenes available include:

Amsterdam - Schipol
Atlanta - Hartsfield
Boston - Logan
Chicago - O'Hare and Midway
Dallas - Love Field
Denver - Stapleton
London - Heathrow
Los Angeles International
Manchester, England
Miami International
Minneapolis, Minnesota
New York - Kennedy and LaGuardia
Pittsburgh, Pennsylvania
Prestwick, Scotland
Sacramento, California
Salt Lake City International
San Francisco International
Seattle-Tacoma, Washington
Stockton, California

AIR TRAFFIC CONTROL SIMULATOR

The Air Traffic Control (ATC) simulator is primarily intended to enhance the realism of the missions for the simulator flight crews. The ATC environment is a significant contributor to pilot workload, and therefore to the performance, of crews in flight. The usefulness of a full-mission simulator is therefore greatly affected by the degree of realism of the ATC model.

The ATC system provides the MVSRF with the capability of simulating the multi-aircraft, multi-ATC environment required to perform full-mission flight simulation. The ATC system is capable of simulating four air traffic controller positions, four "pseudo-pilot" stations, multi-channel voice disguising between the lab and the simulators, and frequency assignment for each ATC sector. The MVSRF ATC environment was developed and upgraded by the Massachusetts Institute of Technology. From the crews' standpoint, this environment consists of dynamically changing verbal or data-link messages, some addressed to or generated by them, others addressed to or generated by other aircraft flying in the immediate vicinity.

The dynamic nature of the ATC is achieved through the use of "operators" manning the ATC station, and controlling both the test aircrew and a number of other aircrew whose actions may interfere with the test crew's actions. Controllability and repeatability of an experiment is achieved by an ATC script mechanism which allows the experimenter to exercise varying degrees of control over the performance of the "operators" as well as the timing of simulation events. Research staff can script scenarios including verbal dialog between controller and pseudo-pilots, specific clearances issued from each controller position, change in weather, winds, and/or ATC information, and visual and non-visual conflicting traffic.

In order to provide a maximum of flexibility in the scope of experiments as well as in their preparation, the ATC simulator is capable of operating in four modes:

1. Stand-alone, without required participation by the rest of the facility;
2. Single-cab mode, with either the ACFS or 727 cab actively participating in the study;
3. Dual-cab mode, with both cabs actively participating, and;
4. Dual experiment mode, with both cabs running independent experiments using the ATC simulation simultaneously.

To increase simulation flexibility, the ATC simulator provides four independent controller stations. These stations can be reconfigured from one simulated geographical area to another in less than one minute, allowing the simulated control positions to "leapfrog" in order to follow the progress of a mission. Single-controller operation is also possible for simpler missions.

The training required of the ATC operators has been simplified by three means: first, the format of the displays has been modified from that of the current National Airspace System (NAS) and Automated Radar Terminal System (ARTS) to be oriented toward the needs of an experimenter rather than toward an experienced professional controller; second, automated aids have been added to assist the operator in the controlling task; and third, a script mechanism is available to assist all the operators, ATC and pseudo-pilots with the "canned" or static part of the simulation scenario. Four pseudo-pilot stations are provided, each able to control up to 25 pseudo-aircraft simultaneously, for a maximum total traffic flow of 100 aircraft in any simulation. A picture of the ATC simulator in operational use is shown in Figure 4.

EXPERIMENTER INTERFACE

Most investigators wish to observe, record, and where possible, quantify many aspects of flight crew behavior during human factors studies in the MVSRF. The facility permits a fine-grained evaluation of individual and crew performance under a wide variety of circumstances. A data acquisition system is provided which enables an observer to document his own observations. It is capable of being tailored to meet the special needs of individual investigators, however, a core of basic data is always obtained in order to build a MVSRF database on flight crew behavior in full mission simulations.

The experimenter interface is flexible enough to readily permit its modification as our knowledge of performance assessment in line-oriented simulation increases. The data collection subroutines can be altered as necessary to achieve the desired processed results. It is anticipated that most of the basic data collected at the MVSRF will be useful for a considerable period of time and that experimenters will want to have repeated access to it.

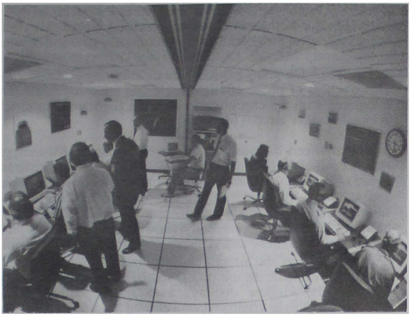


Figure 4 View of the ATC Simulation in operations

EXPERIMENTER STATIONS

Two experimenter stations are provided in the MVSRF control room, one for each of the two flight simulators. They are co-located and can be reconfigured to support future experiments involving simultaneous operation of both flight simulators. Each experimenter station contains computer graphics display systems, keyboards and terminals for interacting with the simulation computers, status lights and emergency controls, communication systems, and other equipment useful or necessary for controlling the flight simulators and conducting simulation experiments.

Each experimenter's laboratory also contains an audio station so that experimenters may communicate with the simulator flight crews during an experiment or with observers located "on-board". The experimenters can also act as air traffic controllers or as pilots of other "pseudo-aircraft" to reduce the number of personnel required to conduct certain simulations. In addition to the main experimenter consoles, an experimenter (or observer) station is located aboard each of the flight simulators. These stations are equipped with a graphics system, keyboard, and other controls and communicators, and can be used to perform most of the same functions as the control room stations. It is also possible to communicate with the Air Traffic Control simulator from each of the experimenter stations. A picture of an experimenter/operator station is depicted in Figure 5.

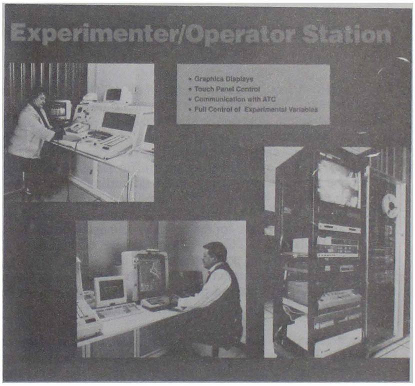
DATA COLLECTION AND ANALYSIS

Data collection within the MVSRF is very flexible. Each of the flight simulator data variables within a "global data pool" may be sampled according to the needs of the experiment or simulation. While there exists an upper limit to the total amount of data which may be sampled and stored, this limit is large. The typical practice is to sample and store a selected number of variables, each of which is time coded to facilitate later analysis. Following completion of a simulation, this data file is available for post-experimental analysis.

The ATC simulator maintains a separate data collection capability. The ATC simulation is also time-stamped and may be merged with the data from the piloted simulators following completion of a given simulation or experiment. As noted above, selected variables may be monitored in real-time during the course of a simulation. It is also possible to record time-coded communications between simulator crew members and ATC personnel (or experimenters) for correllation with the objective flight data.

Other measurements of crew behavior and performance can be effected as experimental needs dictate. Closed circuit TV monitoring and recording is possible, if required. Direct observation and monitoring of flight crew behavior and performance is also possible from within each simulator cab through the use of the experiment/observer station. Data thus recorded can be integrated with the remaining simulation data. A "replay" capability for each simulator allows limited recreation of portions of simulated flights or ATC scenarios when desired. The simulations can also be "frozen" at selected points within the simulation if required by the experimenter.

Post-experimental and post-simulation data analysis may be performed within the MVSRF. A variety of standard statistical and other analysis packages are available. Special analyses, of course, may also be developed on an individual basis. (A very limited amount of real-time data analysis may be possible during some simulation experiments).



Experimenter/Operator Station

- Graphics Displays
- Touch Panel Control
- Communication with ATC
- Full Control of Experimental Variables

Figure 5 Experimenter/Operator Station & Equipment

COMPLETED RESEARCH PROJECTS

Since activation in 1984, approximately 40 complex human factors simulation studies have been conducted at the MVSRF for the FAA, airframe manufacturers, airlines, universities, and NASA. These studies have included evaluation of major operational aviation systems associated with the National Airspace System. Typical studies have included the following:

Traffic Alert and Collision-Avoidance System (TCAS II): This study evaluated the use of a traffic collision avoidance system to prevent the possibility of a midair collision. As a result of this study the FAA has mandated that all large commercial airliners install TCAS in their aircraft by the end of 1993.[1] NASA Ames was solely responsible for the human factors evaluation of TCAS II prior to the airlines' in-service evaluation. NASA studies at the MVSRF verified that pilots can accurately and quickly use TCAS II information to avoid serious traffic conflicts without a significant increase in crew workload.

Microwave Landing System (MLS): This study evaluated the use of a Microwave Landing System as a candidate to replace the current Instrument Landing System (ILS) and to become the standard precision guidance system used for final approaches and landings at major national and international airports. This FAA sponsored program used actual New York metropolitan area air traffic controllers and demonstrated that curved approach paths flown with MLS can increase traffic flow significantly under instrument weather conditions. Results, to date have indicated an increase of 3-20 more landings per hour and could be crucial in the go-no go decision on whether or not to implement MLS.[2]

FAA/USAF/Industry Workload Study for Cockpit Certification: This two year research program sponsored by the USAF and the FAA was directed toward the evaluation of crew workload techniques for aircraft certification. This study was conducted at the MVSRF as a joint effort between the two major U.S. manufacturers of commercial transport airplanes, Douglas Aircraft Company and Boeing Commercial Airplanes. This study provided assessment criteria to enable the FAA to evaluate workload measurement plans for crew size substantiation and workload acceptability during aircraft certification efforts.[3] The results and the lessons learned from this study drove the test plan for crew complement evaluation of the Douglas MD11 aircraft and are being used by Douglas and the USAF for developing the crew complement evaluation plan for the C-17 aircraft.

Air Traffic Flow Management and ATC Automation Experiments: This study is an element of the Aviation Safety/Automation Program initiated several years ago, and is aimed at evaluating an integrated set of automation tools designed to assist air traffic controllers. The objectives of this study were aimed at optimizing air traffic flow to minimize delays and fuel consumption without degrading safety in the National Airspace System. The combined facilities of the MVSRF and the NASA Ames ATC Automation Laboratory provide a unique resource for the development and evaluation of controller tools and ATC procedures where both pilot and controller behavior and performance can be studied in a high fidelity research environment. With a high level of enthusiasm evident among participating pilots, results show increases in landing rate and decreases in controller workload.[4]

Terminal Weather Information Management Experiment (TWIME): This study was part of the Aircraft Environment Management Program and a sub-element of the Aviation Safety/Automation Program. This study evaluated an advanced cockpit weather display presenting ground-to-air data-linked weather information overlaid onto an airport map display. Results of this study showed that pilots could easily avoid severe weather associated with microburst windshears when real-time thunderstorm information is available to the flight crew.

Smart Electronic Checklists: This study is a sub-element of the Aviation Safety/Automation Program designed to develop and demonstrate concepts of error tolerant flight and cockpit management in highly automated aircraft. This study evaluated the usefulness of electronic checklists with various levels of sophistication and intelligence for performing normal procedural tasks while dealing with onboard malfunctions. Three levels of checklists using touch sensitive menu screens for operation of aircraft systems were compared, the first being an electronic version of a standard paper checklist which simply marked off checked items; the second level expanded on the first by sensing system state and indicated to the flight crew if actions had been performed; the third level continuously "autosensed" system state of aircraft variables to indicate to the flight crew current status of aircraft system parameters. Results indicated that subjected pilots were highly in favor of the use of electronic checklists. Activation or selection of checklists items automatically displayed the corresponding aircraft schematic or system display providing an easy-to-use system, thereby reducing pilot workload. However, results indicated that pilots relied heavily on the checklists to check system state and became complacent, instead of performing the system checks themselves.[5] On-going experiments are currently evaluating the effectiveness of phase-of-flight logic and checklist reminders integrated with the checklists to alert the flight crew to check skipped items or aircraft functions.

3-D Spatialized Sounds for Enhanced Situation Awareness: Another sub-element of the Aviation Safety/Automation Program aimed at improving auditory delivery of cockpit communications and warnings by utilizing three dimensional localized stereo sound techniques implemented through the flight crews headsets. Results of this study showed an increase in pilot situational awareness and reduced response time to possible mid-air collisions and aircraft malfunctions.[6] On-going studies will further evaluate the use of 3-D spatialized sounds integrated with out-the-window displays as a traffic collision and avoidance tool to be compared with current head down displays.

Data Link: A joint FAA-NASA study aimed at finding ways to reduce the number of incidents due to communications problems between flight crews and air traffic controllers. This study evaluated a graphical pilot interface to data-link to determine the effectiveness of digitized data-link communications versus conventional audio transmissions, and to see how the use of data-link affected the flight crews situation awareness and ability to fly the airplane despite the loss of "party-line" information obtained by monitoring different radio frequencies. Results of this study will help in the future design and development of guidelines and procedures for the implementation of digital information transfer in future commercial transport airplanes.

Integrated Airport Map Displays: This study evaluated the usefulness of an integrated airport map display which depicted other ground traffic as well as ground clearances data-linked from air traffic control. It is hypothesized that integrated taxi displays will help increase pilot situational awareness about airport terminal areas. Results of this study should help to avoid collisions or incidents such as the ones which recently took place at Detroit and Los Angeles.

PLANNED UPGRADES

When the MVSRF was built in the early 1980's, the 727-200 airplane was the most common airplane flying at the time. With the current changes in technology and emphasis towards a "glass cockpit environment" in the airline industry, it was decided to replace the Boeing 727-200 simulator with a representative current technology glass cockpit simulator capable of addressing the types of research issues that need to be addressed in the years to come. A CAE built Boeing 747-400 simulator, equipped with a McDonnell Douglas Vital VIIe daylight/night/dusk visual system, reprogrammable flight displays and avionics, a fully digital control loading and motion system, a digital sound and aural cues system, and a weather radar simulation will be installed at the MVSRF in autumn of 1993. The simulator will be certified to FAA Level D training standards in accordance with FAR part 121 Appendix H and FAA Advisory Circular AC120-40B. The 747-400 simulation and data collection systems software will be hosted on two IBM 6000 reduced instruction set computers (RISC). New ideas or concepts may then be examined or studied first in the Advanced Concepts Flight Simulator to test the validity of a new type of system or design, and then tested in the 747-400 simulator to verify the acceptance of such a system in an actual airplane cockpit representative of the current and future operating fleet of aircraft.

Upgrades planned for the ACFS include the integration and evaluation of a graphical interface for the ACFS flight management system, the integration of intelligent systems that will track and model pilot behavior, and the integration of various models of fault monitoring and diagnostic systems that will detect, diagnose and predict the effects of aircraft malfunctions or failures and how they may lead to dangerous situations.

SUMMARY

The MVSRF has proved to be a vital tool for the study of human factors research. Its unique research capabilities enable scientists to develop and test new concepts in a realistic cockpit environment through the use of full-mission simulation. This facility enables researchers to conduct studies of how crew members interact with each other and with the machines aboard the flight deck. The facility's role is of primary importance in our continuing efforts to reduce the incidence of human error and improve the overall efficiency of the National Aviation System. The MVSRF will continue to make an impact supporting programs such as the Aviation Safety/Automation Program, the National Plan for Human Factors, the Terminal Area Productivity Program and the High Speed Civil Transport.

References

1. Chappell, S., Billings, C., Scott, B., Tuttell, R., Olsen, M., & Kozon, T. Pilot's Use of a Traffic Collision and Avoidance System (TCAS II) in Simulated Air Carrier Operations - Vol. I: Methodology, Summary and Conclusions. NASA Technical Memorandum 100094, 1989.

2. Scott, B., Dargue, J., Goka, T. Evaluation of Advanced MLS Procedures in the New York Terminal Area. Prepared for FAA MLS Program Office (AND-30), 1991.

3. Corwin,W., Sandry-Garza, M., Biferno, M., Boucek, G., Logan, A., Jonsson, J. & Metalis, S. Assessment of Crew Workload Measurement Methods, Techniques and Procedures, Vol. I - Process, Methods and Results. Wright Research and Development Center Report WRDC-TR-89-7006, September, 1989.

4. Davis, T. & Green, S. Piloted Simulation of A Ground Based Time-Control Concept for Air Traffic Control. AIAA Guidance and Control Conference, Boston, Mass., August 1989.

5. Palmer, E. & Degani, A. Electronic Checklists: Evaluation of Two Levels of Automation. Presented at the 6th Symposium on Aviation Psychology, Columbus, Ohio, April 29 - May 2, 1991.

6. Begault, D.R. A Head-Up Auditory Display for TCAS Advisories. Submitted to Journal of Human Factors Society, 1992.

AIAA-92-4178-CP

PILOTED EVALUATION OF AN INTEGRATED PROPULSION AND FLIGHT CONTROL SIMULATOR

Michelle M. Bright
NASA Lewis Research Center
Advanced Controls Technology Branch
Cleveland, OH 44135

Donald L. Simon
US Army Aviation Systems Command
Propulsion Directorate
Lewis Research Center
Cleveland, OH 44135

ABSTRACT

This paper describes a piloted evaluation of the integrated flight and propulsion control simulator at NASA Lewis Research Center. The purpose of this evaluation is to demonstrate the suitability and effectiveness of this fixed base simulator for advanced integrated propulsion and airframe control design. The evaluation will cover control effector gains and deadbands, control effectiveness and control authority, and heads up display functionality. For this evaluation the flight simulator is configured for transition flight using an advanced Short Take-Off and_Vertical Landing fighter aircraft model, a simplified high-bypass turbofan engine model, fighter cockpit, displays, and pilot effectors. The paper describes the piloted tasks used for rating displays and control effector gains. Pilot comments and simulation results confirm that the display symbology and control gains are very adequate for the transition flight task. Additionally, it is demonstrated that this small-scale, fixed base flight simulator facility can adequately perform a real time, piloted control evaluation.

INTRODUCTION

The Advanced Controls Technology Branch at NASA Lewis is conducting research in the area of integrated flight and propulsion control design, specifically for a Short Take-Off Vertical Landing (STOVL) aircraft. The flight simulator facility was developed to provide a means to validate integrated design methodologies, to monitor engine and airframe parameters during real time simulation, to evaluate new software partitioning methods, and to test control specification bandwidths and control rates through piloted engineering evaluations. This flight simulator has undergone evaluation by certified test pilots for maneuverability, controllability, and basic functionality to prove that it is a credible, realistic, real-time simulator. This paper describes the evaluation of this flight simulation environment with a brief description of the actual test environment; the control design and physics models used to test the real time capabilities of the simulator; the cockpit effectors and displays used for this evaluation; and the flight scenarios and profiles used for the piloted testing of the flight simulator. Finally, piloted comments and conclusions concerning the suitability of the flight simulator for current research, and recommendations for enhancements are given.

SIMULATION TEST ENVIRONMENT

The flight simulator facility, as shown in Figure 1, consists of an image generation system and UNIX development station, a mockup fighter cockpit, a real time simulation computer, and a control computer system. The image generation system generates the Heads Up Display (HUD), the Heads Down Display (HDD), and the out-the-window scenery using 3 video channels to provide 150 degrees field of view. The fighter cockpit provides pilot effectors for the control of engine and airframe commands. The real time simulation computer executes the real time engine and airframe physics models. Finally, the control computer system executes the integrated control design algorithms. A complete description of this simulation facility is given in reference [1].

For the evaluation of this flight simulator, sample aircraft and engine models, and control designs were selected to test its capabilities. This evaluation had several purposes. First, the minimal heads up and heads down display symbologies required to perform the sample control task were determined. When the flight simulator configuration could not accommodate the predefined displays as defined in reference [2]; pilot rated, acceptable alternatives that serve the same function were developed. Also, the minimal gains and deadbands for the flight simulator cockpit effectors were determined. In addition, time delays in the simulator response time were measured, and the out the window sceneries were judged for effectiveness during the piloted control task.

CONTROL DESIGN AND PHYSICS MODELS

The vehicle model for this simulation test is a six degree of freedom, delta winged E7-D aircraft with a multi-nozzle turbofan engine shown in Figure 2. The airframe is configured with an ejector nozzle, a ventral nozzle, a 2-dimensional convergent/divergent aft nozzle, and a Reaction Control System (RCS). The RCS allows for control of aircraft attitude during hovering flight. The engine for this aircraft is a mixed flow, vectored-thrust configuration. For this investigation the integrated engine and airframe equations of motion are 14th order with 12 inputs and 10 outputs, and represent a linear, simplified model. Further information about the vehicle, the airframe model, and the engine model can be found in reference [3].

The integrated flight and propulsion controller used for this experiment is a reduced order H-infinity design, which is a linear, 21st order system. The controller includes limiting logic and fan speed scheduling, and is configured only for the transition phase of flight from cruise to hover. A detailed description of the control design is found in references [4,5].

Figure 3 displays a high level view of the discrete linear control design used for this experiment. The pilot inputs from the cockpit effectors are sent to the controller and are scaled by the input effector gradients and prefiltered for command shaping and blending. The prefilter and control blending convert the pilot selections of acceleration, pitch rate, flight path angle, roll rate, and sideslip, into desired velocities, accelerations, and body angles or rates for the controller. The original values for the prefilters were based upon desired handling quality characteristics of the E7-D aircraft during piloted simulations at NASA Ames Research Center and cockpit configuration tests at General Dynamics, Fort Worth Division. Based on a review of these efforts, it was determined that the NASA Lewis flight simulator could not exactly replicate the implementation of these control effectors. Therefore, the pilot gradients were modified to reflect a displacement control stick, instead of a fixed force sidestick controller as used in the General Dynamics study. The throttle displacement gradients also were modified to reflect linear displacement rather than angular displacement. Further information on the implementation of these control modes for a STOVL task are described in reference [7].

COCKPIT EFFECTORS AND DISPLAYS

Development of the Pilot Vehicle Interfaces (PVI) for this flight simulator was based upon PVI research by Merrick, Farris, and Vanags at NASA Ames Research Center [2]. For demonstration purposes, a STOVL aircraft model, which is described below, was chosen with its associated HUD symbology, HDD instrumentation, and cockpit effector configuration.

The HUD symbology was generated and updated on the visual system development station. The displays and scenery were modified to reflect an integrated engine and airframe control task, typical of a STOVL aircraft. Figure 4 shows an example HUD symbology which was implemented on the flight simulator. The symbology includes a pitch ladder, heading scale, aircraft reference symbol, and flight path symbol. Additionally, engine and aircraft parameters such as altitude, airspeed, forward acceleration, and vertical acceleration rates also are displayed. This symbology was pilot rated during the flight evaluation, and the throttle position and thumbwheel position symbols were added due to pilot preference. A further discussion of the pilot ratings is given in the results section of this paper.

The switches and effectors in the mock-up fighter cockpit are implemented to reflect the simulation of an integrated flight and propulsion control task. The cockpit effectors were based upon a "rate system" command structure. This rate system was implemented to accommodate the three modes of flight that the example STOVL aircraft can encounter: cruise, transition, and hover. With the rate system commands, the longitudinal stick provides pitch rate/attitude hold; the lateral stick provides roll rate/bank angle hold; the rudder pedals provided sideslip commands; and the linear throttle commands flight path angle.

An additional control effector and a digital switch were added for this simulation -- a rotating thumbwheel and a reset switch. The thumbwheel, positioned on the linear throttle, commands acceleration/deceleration during the transition to hover flight regime. The reset switch, which is normally the trigger switch of the sidestick controller, toggles the simulation between initial condition mode and operate mode. If the simulation reaches saturation limits of the control actuators, the simulation will automatically reset to the initial condition mode. The trigger/reset button places the simulation back into operate mode. A diagram of the cockpit effectors and their functionality is found in Figure 5.

EVALUATION TASKS AND PROCEDURES

To evaluate the control gains and bandwidths for each of the control effectors, the fixed base simulation piloted tasks included the following: (1) straight and level flight to evaluate pitch, (2) straight and level flight to evaluate roll, (3) curved decelerating runway acquisition to evaluate roll and pitch harmony, (4) decelerating approach to runway at various airspeeds to evaluate acceleration/deceleration performance, and (5) decelerating approach to runway and then accelerating to cruise while varying flightpath angle to evaluate flight path response. All scenarios are performed with the aircraft configured for transition phase of flight. The scenarios begin at 1000 feet altitude, 120 knots airspeed in the landing configuration. The aircraft's initial position is changed to either the right or left side of the runway with a 600 foot offset at 4.5 miles away from the final landing point for the curved approaches [6].

The first task was to evaluate the pilot gradients and deadbands associated with the longitudinal control stick. For this evaluation the straight and level flight was performed by the pilot. The sequence of events for this task was to acquire a 5 degree pitch angle, level out, and then acquire a -5 degree pitch angle. This sequence was increased to 10 degrees pitch and the order of the task was reversed.

The second task was to evaluate the pilot gradients and deadbands associated with the lateral control stick. Straight and level flight also was performed for this test. The sequence of events for this task was to acquire a 10 degree bank angle, level out, and then acquire a -10 degree bank angle. This sequence was increased to 30 degrees bank and the order of the task was reversed.

The next task was to evaluate the pilot gradients and deadbands associated with the lateral and longitudinal stick blending. For this evaluation the curved decelerating runway acquisition task was performed by the pilot. The sequence of events for this task was decelerate at 0.1g along a -3 degree flightpath, then bank right or left to align with the final approach course and maintain airspeed and altitude above the runway. While above the landing site, small pitch and roll adjustments were made to remain aligned with the runway. The initial position of the aircraft was 1000 feet altitude and 4.5 miles from the landing point. For a more difficult tracking task, the distance from the runway was decreased to 3.0 miles and then to 1.5 miles. In this manner, the sharp turning task provided information on the combination of pitch and roll necessary to acquire the runway and maintain alignment.

Another task was to evaluate the pilot gradients and deadbands associated with the thumbwheelcontrolling acceleration/deceleration. For this evaluation the decelerating approach to the runway at various airspeeds was performed by the pilot. The sequence of events for this task was to acquire a 0.1g rate of deceleration along a -3 degree flightpath, then maintain altitude above the runway at an airspeed of 100 knots, 80 knots, and then 60 knots. This task was repeated for 0.2g deceleration.

The last task was to evaluate the pilot gradients and deadbands associated with the throttle controlling rate of climb or descent. For this evaluation the decelerating approach was performed, followed by the accelerating transition to straight and level flight (wave-off). The sequence of events for this task was to commence a rate of deceleration of 0.1g along a -3 degree flightpath and a -6 degree flightpath, acquire the runway, and maintain airspeed at 80 knots, at 50 feet above the runway. Then, the pilot accelerated to above 95 knots at 0.5 g and then acquired a 3 degree and 5 degree flightpath angle to cruise in conventional flight. This scenario was repeated for a 0.2g deceleration and -6 degree flightpath for a more difficult flightpath control task.

PILOT COMMENTS AND RESULTS

For the pitch control task the pilot found the longitudinal stick responded sluggishly with a considerable time lag between command and aircraft response. This indicated that the deadband of the prefilter was too large for both the small and large pitching task, thus, the deadbands and gains were modified and the tasks were repeated. In this second test the longitudinal stick responded crisply, without much pilot effort for both the small pitching task and the large pitching task in the transition flight mode. The pitch ladder on the HUD responded properly, without noticeable time lag, and the overall rating was good. For this task the original gains and deadbands and the pilot preferred gains for the longitudinal stick appear in figure 6.

For the roll control task the pilot found considerable time lag in the roll response and found that the task required substantial movement in the control effectors. There was no pitch and roll gain harmony between the longitudinal stick and the lateral stick. Various gains and deadbands were tried for the more difficult runway acquisition tasks, but the hardware did not perform adequately, and the pilot continued to make large adjustments to compensate for the poor response of the lateral stick. It was determined at this time that a drift problem existed in the roll axis because the lateral stick had some "slack" and did not always return to center. Due to this problem the minimal deadbands for this control effector were examined.

To resolve the deadband problem in the roll controller, a simple experiment was performed to ascertain when the "slack" in the stick caused a perceptible roll command. During the simulation a small pressure was applied to the roll controller in each direction of the "slack". The magnitude of the upper and lower deadbands was decreased while the roll command was monitored. Once the roll angle began to drift, the minimum upper deadband was found to be 0.1 inches of deflection, and the minimum lower deadband was found to be 0.06 inches of deflection. The results of this experiment are shown in Figures 7 and 8.

Using this information, the deadbands were set to this minimal "drift limit", and the roll control task and the curved decelerating runway acquisition tasks were repeated to evaluate the lateral control stick performance and the pitch/roll gain harmony. The original gains and the improved effector gains are given in figure 9. These gains reflect good pitch and roll harmony, good handling qualities of the control effectors for this task, and appropriate operation of the heads up display symbology. Additionally, the 150 degrees field of view scenery was a large improvement over the original single channel system and was a necessary expansion in order to perform the runway acquisition task.

The next task to evaluate acceleration/deceleration originally had been implemented using the throttle effector in the General Dynamics study.[7] Due to pilot preference, this study concluded that the thumbwheel should control acceleration/ deceleration. Because of this new implementation, the thumbwheel gains were scaled to reflect the change in effectors. Initially, this effector was also difficult to accurately evaluate because there was no detent to show null position, and the thumbwheel could rotate 270 degrees. This was not acceptable for this evaluation, therefore, the rotation was limited to 36 degrees. (approximately the span one's thumb could move in a single motion). Also, no other hardware modifications could be made on this effector because it will serve a different function in the cruise and hover modes, therefore, a heads up display symbology was used to show thumbwheel position. (please refer to figure 4.)

Once these modifications in hardware were implemented, the precision task of decelerating to the runway was evaluated. The pilot found the scaled thumbwheel gains were very crisp and the symbology responded very sharply.

Consequently, the original gains were altered only to reflect the 36 degree rotation limit and the scaling due to the change in implementation. These gains appear in figure 10.

The final evaluation was of the throttle effector. The throttle had originally controlled acceleration/deceleration, however, the pilot preferred to control rate of descent with this effector. Because of this new implementation, the throttle gains were scaled to reflect the change in effectors. Initially, this effector was also difficult to accurately evaluate because there was no detent. The heads up display, once again, provided information to display throttle position. Once this was implemented, the tasks of decelerating to the runway and accelerating to cruise were performed. The pilot found the throttle response was very crisp, with no response delays in the flightpath symbol. The original gains and the new gains used for this task are presented in figure 11.

The rudder pedals gradients and deadbands were not evaluated because of their limited use during the transition case scenarios, however, they were evaluated to show proper functionality and response to pilot commands. The default pilot gradients used for this evaluation are given in figure 12.

CONCLUSIONS

The integrated propulsion and flight simulator has successfully been designed, built, and demonstrated as a real time, pilot-in-the-loop, evaluation station for integrated engine and airframe control laws. The flight simulator performed very adequately in the piloted ratings for the fixed base simulation of a Short Take-Off Vertical Landing aircraft during the transition to hover phase of flight. Control effector gradients, deadbands, and heads up display symbologies were evaluated by the pilot. The flight simulator was configured to reflect pilot preferences of the control effectors and displays. Piloted ratings show that this fixed base flight simulator configuration, with its associated control effector gradients, gains, and displays, can be used to adequately evaluate integrated flight and propulsion control laws in transition to hover scenarios. As a further demonstration of this simulator's capability, a full control evaluation of the transition to hover case scenario will be performed on the fully nonlinear STOVL aircraft model, engine model, and complete control design.

ACKNOWLEDGEMENTS

The authors express their sincere appreciation to Richard Ranaudo, the NASA Lewis Pilot, for his expertise during this investigation. Additionally, we extend our thanks to John DeLaat, Sanjay Garg, and Duane Mattern for their technical assistance.

REFERENCES

1. Bright, Simon, "The NASA Lewis Integrated Propulsion and Flight Control Simulator", NASA TM-105147 and Proceeding of the AIAA Flight Simulation Technologies Conference, New Orleans, La., August 12-14, 1991.

2. Merrick, Farris, Vanags, "A Head Up Display for Application to V/STOL Aircraft Approach and Landing", NASA TM-102216, 1990.

3. Akhter, Vincent, Berg, Bodden, "Simulation Development for US/Canada Controls Technology Program", Proceedings of the Twentieth Annual Modeling and Simulation Conference, University of Pittsburgh, Pittsburgh, PA. June 1989.

4. Garg, Mattern, Bright, Ouzts, "H-Infinity Based Integrated Flight Propulsion Control Design for a STOVL Aircraft in Transition Flight", NASA TM-103198 and Proceedings of the AIAA Guidance, Navigation and Control Conference, Portland, Oregon, August 20-22, 1990.

5. Garg, Mattern, "Application of an Integrated Flight/Propulsion Control Design Methodology to a STOVL Aircraft", NASA TM-105254 and Proceedings of the AIAA Guidance, Navigation and Control Conference, New Orleans, La., August 12-14, 1991.

6. Franklin, Stortz, Engelland, Hardy, Martin, "Moving Base Simulation Evaluation of Control System Concepts and Design Criteria for STOVL Aircraft", NASA TM-103843, June 1991.

7. Whatley, Virnig, Bodden, "Implementation of STOVL Task Tailored Control Modes in a Fighter Cockpit", AIAA/AHS/ASEE Aircraft Design, Systems and Operations Conference, Dayton, Ohio, September 17-19, 1990.

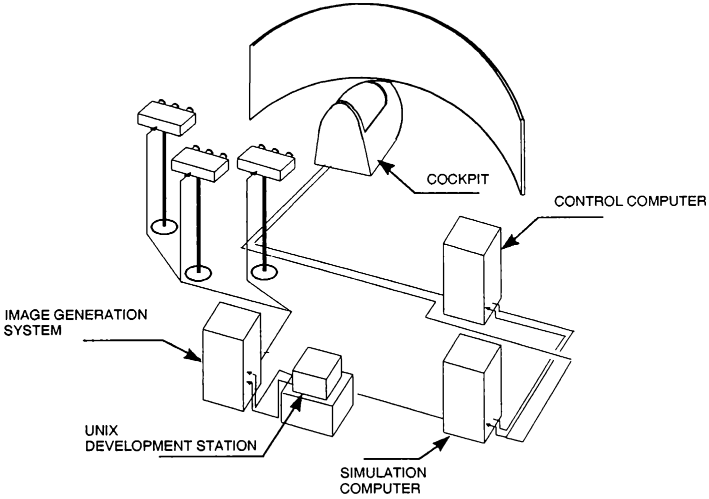


Figure 1. Flight Simulator Facility

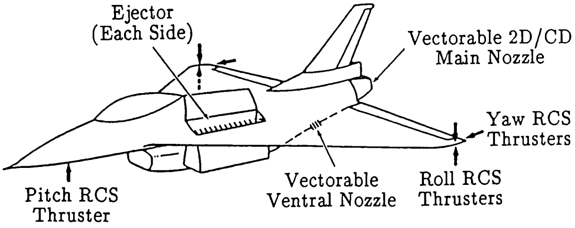


Figure 2. E7-D Aircraft Model

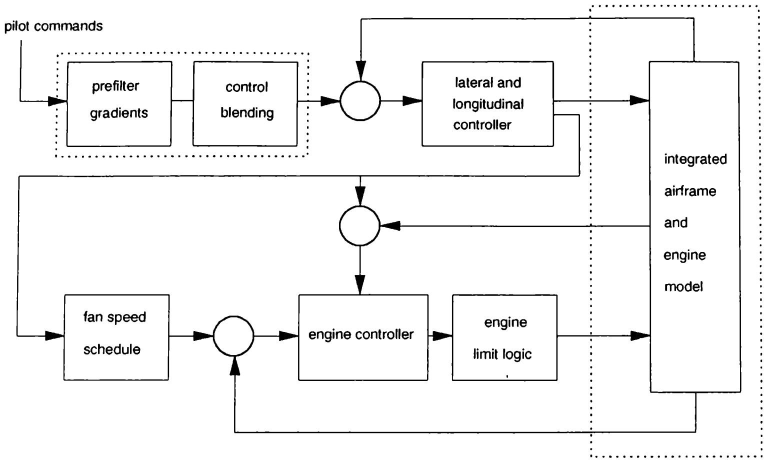


Figure 3. Block Diagram of Transition Control Design

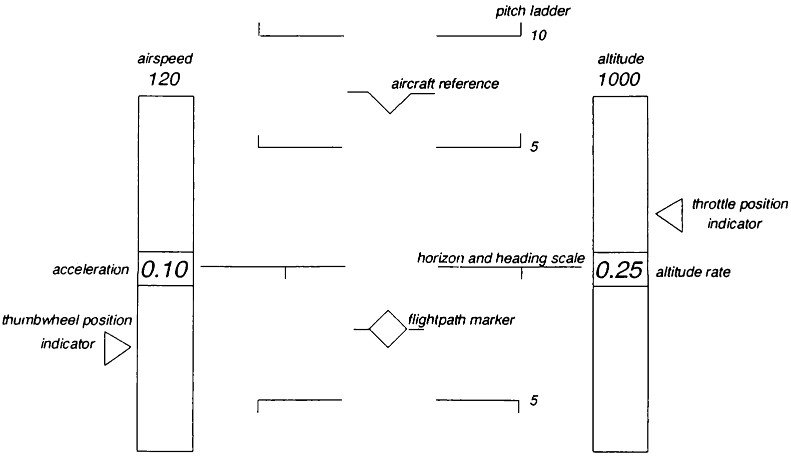


Figure 4. Heads Up Display Symbology for Transition Flight

control inceptor pilot command	Throttle	Thumb Wheel	Longitudinal Stick	Lateral Stick	Rudder Pedals
accel/decel		X			
flightpath angle	X				
rollrate				X	
pitchrate			X		
sideslip					X

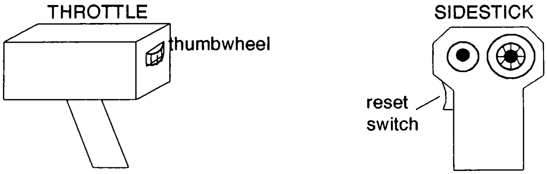


Figure 5. Cockpit Effectors in Transition Flight Mode

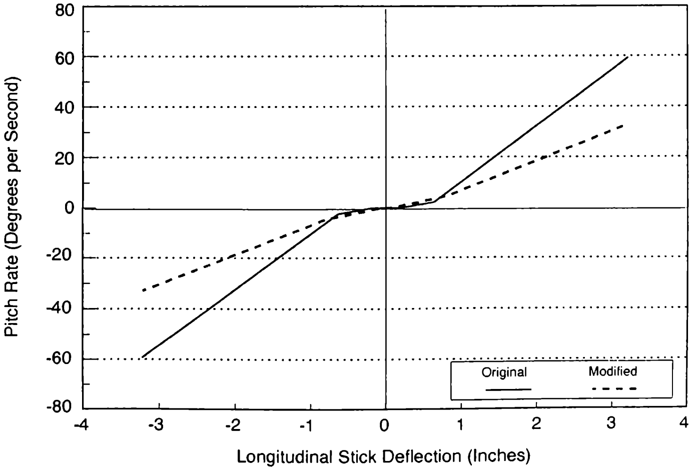


Figure 6. Longitudinal Stick Gain and Deadband

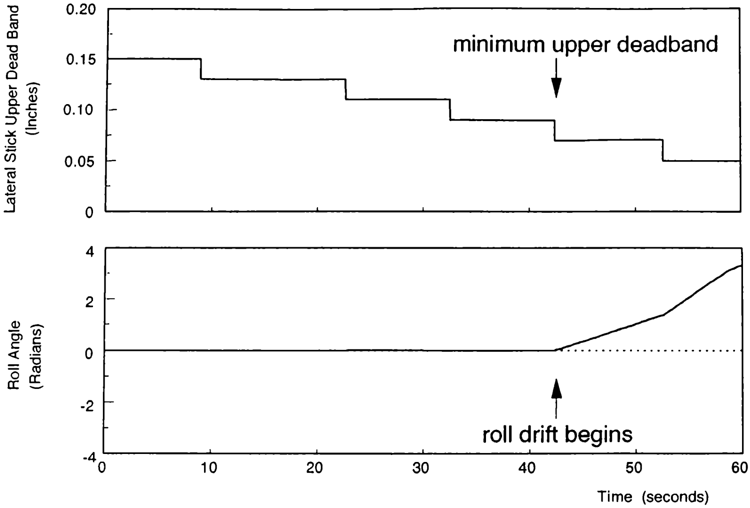


Figure 7. Upper Deadband for Lateral Stick

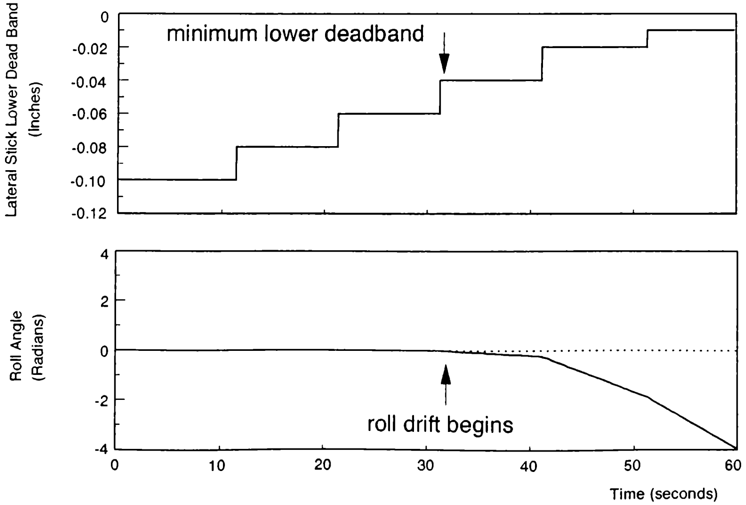


Figure 8. Lower Deadband for Lateral Stick

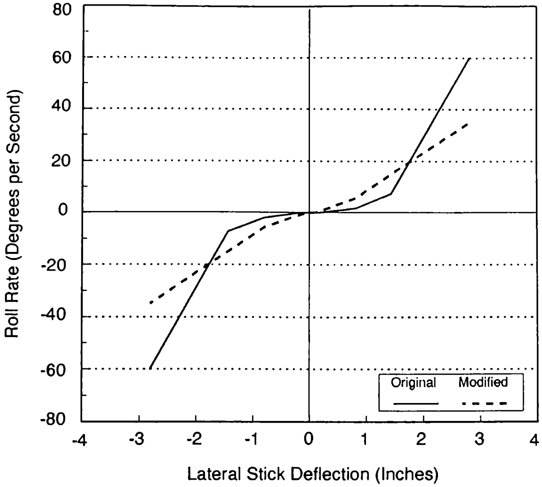


Figure 9. Lateral Stick Gain and Deadband

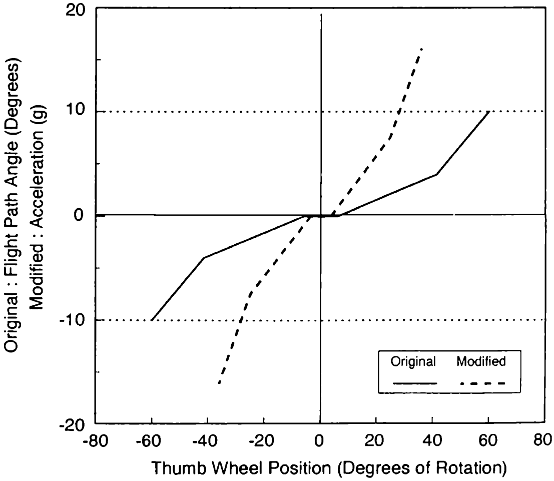


Figure 10. Rotating Thumb Wheel Gain and Deadband

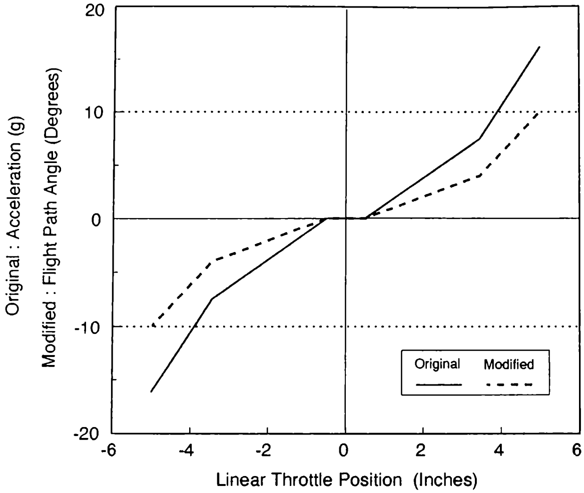


Figure 11. Linear Throttle Gain and Deadband

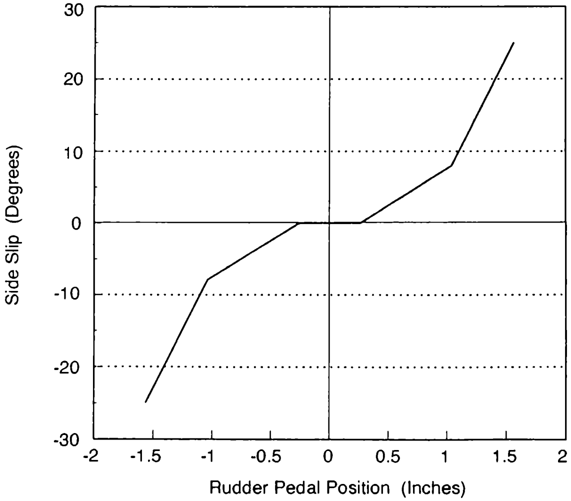


Figure 12. Rudder Pedal Gain and Deadband

AIAA-92-4179-CP

High Performance Flight Simulation at NASA Langley

Jeff I. Cleveland II*, Steven J. Sudik, and Randall D. Grove
NASA Langley Research Center
Hampton, Virginia 23665-5225

NASA's Langley Research Center (LaRC) has been using real-time flight simulation to support aerodynamic, space, and hardware research for over fifty years. In the mid-1960s LaRC pioneered the first practical real-time digital flight simulation system using Control Data Corporation (CDC) 6600 computers. In 1976, the 6600 computers were replaced with CDC CYBER 175 computers. In 1987, the analog-based simulation input/output system was replaced with a high performance fiber optic-based digital network. With the increased complexity and higher performance requirements for the simulation of modern aircraft, LaRC is now replacing the CDC computers with Convex Computer Corporation supercomputers. This paper reviews the hardware and software, experience with the new system, status, and plans.

INTRODUCTION

An unpublished survey of flight simulation users at the NASA Langley Research Center (LaRC) conducted in 1987 projected that computing power requirements would increase by a factor of eight over the coming five-years (Figure 1). Although general growth was indicated, the pacing discipline was the simulation of aircraft flexible modes. Factors influencing growth included: 1) active control of increased flexibility, 2) less static stability requiring more complex automatic attitude control and augmentation, 3) more complex avionics, 4) more sophisticated weapons systems, and 5) the need to simulate multiple aircraft interaction, the so called "n on m" problems. This requirement for more computing power is, if not industry wide, at least common to the high performance aircraft segment.

In 1976, two single processor Control Data Corporation CYBER 175 computers, tightly coupled through extended memory, were installed to support flight simulation. In 1987, a new digital data distribution and signal conversion system, referred to as the Advanced Real-Time Simulation System (ARTSS), was put into service. This system, using the Computer Automated Measurement and Control (CAMAC) technology, replaced two twenty year old analog-based systems. The ARTSS has been very successful and is described in references 1 through 7.

Having decided to continue using centralized computers, LaRC issued a Request for Proposals in May, 1989 and subsequently awarded a contract to Convex Computer Corporation in December of that year. The resulting computational facility provided by this contract is the Flight Simulation Computing System (FSCS). This paper presents the hardware and software comprising the FSCS, experience to date, status, and plans.

CURRENT HARDWARE SYSTEM

The computers that LaRC is putting in place to fulfill the requirements are Convex Computer Corporation C3200 and C3800 series computers. These computers are classified as supercomputers and support both 64- and 32-bit scalar, vector, and parallel processing technology. The first delivery consisted of a Convex C3230 (3 CPUs expandable to 4) with two CAMAC interfaces. The system was delivered with two peripheral buses (PBUS): one PBUS that is used for input/output to standard peripherals such as tape, disk, and line printer and one PBUS that is used exclusively for real-time input/output to the ARTSS CAMAC network. Each VME Input/Output Processor (VIOP) is a Motorola 68020 microcomputer that provides programmable input/output control. Each VIOP is connected to a standard 9U VMEbus and to the corresponding PBUS. The CAMAC interface consists of a KineticSystems Model 2140 Enhanced Serial Highway Driver for VMEbus.

The second delivery consisted of one Convex C3850 (5 CPUs expandable to 8) computer config-

* Senior Member AIAA

ured similar to the C3230 with 2 PBUSs and two CAMAC interfaces. NASA provided two additional interfaces to bring the total to four. The computer contains 512 megabytes of main memory and sufficient disk and other peripherals to support flight simulation. The resulting computer configuration is shown in Figure 2.

LaRC SOFTWARE IMPLEMENTATION

CAMAC Software Driver

Most of the driver software for the CAMAC highway executes in the VMEbus I/O Processor (VIOP). During the real-time phase of the application's execution, all highway I/O is done by interaction between the VIOP and the application. The application has exclusive access to the VIOP and the highway it drives. Responsiveness is enhanced by the VIOP's polling for requests and external events rather than being interrupt driven. Recently the clocking has been generated by an additional card in the VME chassis rather than an event on the highway. The VIOP now needs only reference a directly connected status register rather than doing I/O from the highway to sense the clock. Since access to this card is faster than the computer supplied clock, this card is also used to provide other timing information. This adds greater precision to time critical operations.

Real-Time Supervisor

Real-Time Supervisor, a suite of routines written in C and Fortran, is the run-time library used by the application programs. It provides most of the job's interactions with the system and the real-time environment.

The highway portions define communication areas and buffers, make non-time-critical requests for input or output, retrieve and reformat this data, process and reformat the time-critical I/O, and fetch up-line requests for service from the real-time sites. Recent hardware upgrades have reduced the amount of data reformatting needed thus reducing latency. The DRR (data recording and retrieval) part defines buffers and variable names for recording data, does input, does output, and can reposition the DRR files.

Supervisor gains control on errors and displays a menu giving the user several options, including recovery, analysis, dumping of memory contents and termination. Provisions are included for automatic recovery from frame overruns. The overruns can be accidental as when an unusual logic path takes more computation time that normal; or the overruns can be planned as when the application fills out the frame with non-time-critical operations which operate in the background over several frames. Because of foreground/background capabilities, coding of traceback routines was a significant effort involving some modifications to the kernel.

Site Configuration Management and Scheduling Software

Two utilities, Site Compiler (SC) and Site Linker (SL), are used to maintain site configuration. SC generates machine readable descriptions of simulation sites with some sites used in potentially several different ways. SL links several site descriptions together into a configuration file since most simulations use more than one site. Parts of the configuration file are used by the I/O processor (VIOP), other parts are used by the scheduler, and other parts are used by the application library known as Supervisor. Currently, 1200 different configurations are supported.

The Scheduler is invoked by the user to reserve a highway and configure sites as described in a user selectable configuration file. A highway is configured into a ring network when the scheduler sends an RS232 message to the computer that controls the switch. The switch controller configures any of 12 highways to contain up to 10 of 36 simulation sites. If any element is down or otherwise reserved, scheduling fails. On successful scheduling, the scheduler causes the VIOP to initialize the sites and prepares information that will later be sent to Supervisor. If the current session has a highway already assigned, it will be deallocated before the new one is constructed. Portions of scheduler code also deallocate highways when a user logs out (or hangs up) without deallocating. Other portions deallocate assigned highways during an unscheduled reboot.

PERFORMANCE

CPU Performance

Real-time flight simulation at LaRC requires high scalar CPU performance to solve the equations of motion of the system being simulated. Using an existing X-29 simulation as a benchmark, the following CPU performance was specified:

1. If a single CPU configuration is provided, the CPU must solve the benchmark in at most 165 seconds.
2. If a multiple CPU configuration is provided, each CPU must solve the benchmark in at most 330 seconds.

The CYBER 175 computer solves the benchmark in 660 seconds. The C3230 solves the benchmark in 245 seconds on a single CPU. This yields a performance increase of 2.7 over the CYBER 175. The C3850 solves the benchmark in 120 seconds which yields a performance increase of 5.5 over the

CYBER 175.

Real-Time Input/Output

The ARTSS CAMAC system has provided LaRC with a high performance real-time input/output system that has extended the capabilities of the LaRC simulation system. The transfer rate between the Convex computers and the real-time highways is 24 million bits per second which is the maximum transfer rate of the CAMAC system. See Reference 6.

Responsiveness

One of the critical requirements for any real-time simulation system is system responsiveness. The FSCS system is required to respond to an external event, cause a short FORTRAN program to execute, and post an observable output response, in less than 150 microseconds. This elapsed time, called time-critical system response, is measured at an external port on the computer. The external event occurs at a repetitive rate of 1000 events per second. Both Convex computers provide time critical system response times of less than 35 microseconds. In addition to the time-critical system response, CAMAC input/output response is required to be less than 200 microseconds. CAMAC input/output response is defined as the time between the action of an interrupt generated in a CAMAC crate, transfer of one CAMAC word of data, execution of the short FORTRAN program, and transfer of one CAMAC word of output. CAMAC input/output response has been measured at 152 microseconds on the C3230.

Frame Rate

To support simulation applications needing higher frame rates, LaRC requires the system to run simulations at 1000 frames per second. At this frame rate, during any given frame, the system must deliver at least 600 microseconds of CPU time for the simulation model with 100 bytes of real-time input and 100 bytes of real-time output. The sum of system overhead and real-time input/output must be less than 400 microseconds. The C3230 delivers more than 700 microseconds of model computing time at 1000 frames per second.

SIMULATION APPLICATIONS

The real-time facility regularly supports over forty simulation activities with one-third in production during a week's operation. These simulation projects span the research field from general aviation aircraft to hypersonic aerospacecraft. Real-time simulation applications use one of seven simulators in their research effort, as well as graphic computers for the cockpit displays, computer generated imagery for viewing the outside world, and mini/micro-computers for special effects.

The Differential Maneuvering Simulator (DMS) consists of dual 40-foot projection spheres with cockpits that have out-the-window views, heads-up and other cockpit displays, and other-aircraft target projection systems. The DMS is used to investigate the flight scenarios of modern and prototype high-speed jet aircraft in single pilot, one-on-one, or two-versus-one modes of operation. Using the single pilot mode, NASA and the U.S.Navy are conducting high angle-of-attack simulation studies to develop guidelines for nose-down control requirements. Single air combat simulations incorporate maneuvering logic programs to drive the target aircraft in evasive actions. The other two modes of operation, one-on-one and two-versus-one, aid in developing aircraft maneuverability and agility. New simulation applications are investigating helmet mounted displays for the presentation of aircraft related information and visual displays.

The DMS studies have long been CPU bound and were the first to use the increased CPU power of the initial FSCS computer system. The FSCS permits simulations involving the most sophisticated aircraft in one-on-one and two-versus-one modes of operation. Increased memory capacity permits databases to include all the data required for missions (full mission scenarios) from leaving the hangar, through take off and landing, and the flight objective. The new FSCS offers expanded capabilities for high performance aircraft modeling and sophisticated models.

The Transport Systems Research Vehicle (TSRV) simulator is a duplicate of the highly modified aft flight deck onboard the NASA Boeing 737 aircraft. TSRV studies include investigations of flight crew requirements and operational procedures for a ground-to-air data link air traffic control (ATC) scenarios. Another TSRV application is the development of on-board graphical weather interactive displays from ATC weather information sources.

To support the simulations using the TSRV simulator, the FSCS with its increased CPU power and memory was necessary for the development of the existing programs. The FSCS not only provides the current capability, but will provide future capability for modeling in the areas of guidance algorithms, navigation information, flight path algorithms and data link algorithms.

The Visual Motion Simulator (VMS) is a six-degree-of-freedom motion base platform with a dual seat cockpit fitted with out-the-window visual displays. Studies using the VMS include the simulation of transport aircraft encountering wind shears from microbursts, and high speed civil transport investigations of mach 3 passenger travel over ranges of 6500 nautical miles. The hypersonic aerospacecraft research project studies a powered vehicle designed to takeoff from the ground, accelerate to hypersonic speed, enter earth orbit, re-enter the earth's atmosphere, decelerate and land at an airport. The F-16XL study examines the takeoff, climb-out, and landing/approach flight phases of a modified research aircraft.

The FSCS permits and expands the capabilities for full mission scenarios for high speed civil transport simulation, hypersonic aerospacecraft simulation, and modified research aircraft.

The FSCS offers expanded performance due to its increase in memory capacity and increase in CPU power. The FSCS allows and expands full mission scenarios for TSRV ATC projects, high speed civil transport, hypersonic aerospacecraft, and modern high speed jet aircraft. The FSCS permits high fidelity simulations of weather models, flexible aircraft models and algorithms for guidance, navigation, flight path and data link.

Hardware-in-the-loop simulations include space station structures that are large flexible units with numerous excitation modes. System identification and control law testing is being done with the hardware and sensors in closed-loop and open-loop operation.

The FSCS has the power required to interface with flexible space structures at the required rate of 1000 frames per second. The higher frame rate supports the analysis of the desired solution frequencies while the increased power supports additional excitation modes.

STATUS AND PLANS

The initial FSCS computer (C3230) and an interim computer (C3220) supported the FSCS conversion effort and research production for the real-time application programs. The initial FSCS computer has two CPU's available for real-time operation - two simultaneous real-time programs or one large program using both CPU's. The interim computer handled the secure operations for the real-time application programs and complemented the FSCS computer when not operating in the secure mode. The CDC CYBER computers were used for real-time operation of programs not yet converted, for dynamic checking of programs in the conversion process and for programs that will not be converted.

The simulation application conversion effort to the FSCS computing system has been completed and the respective simulations are in a production status. The DMS application programs were completed first because of the greater need to use the FSCS computing system. The TSRV baseline program was completed first and then the individual project programs were converted according to their time schedule. The TSRV conversion effort was the most difficult since the programs were very large and complicated, and involved special system software.

The final FSCS computer (C3850) is the main computing system for the real-time simulation facility. This system has four CPU's available for real-time operation. This system can be used for four single-CPU simulations or fewer multi-CPU simulations. The interim computer (C3220) was removed when the C3850 entered into production status and the initial FSCS computer (C3230) now handles secure operations and complements the final FSCS computer (C3850) when not running in secure mode. The first real-time CDC CYBER computer was removed in June of this year and the remaining one will be removed later in 1992. Upon completion of their removal, the real-time simulation facility will be operating totally on the new Flight Simulation Computing System.

CONCLUSION

NASA Langley Research Center is nearing completion in the development of a high performance computing system to simulate, in real-time, increasingly complex and high performance modern aircraft. With CPU power gains of over twenty compared with the previous real-time system and with large gains in central memory, research never before feasible is being routinely done. Using centralized supercomputers coupled with a proven real-time network technology provides LaRC with a highly flexible, easily manageable flight simulation system. This system will provide high performance flight simulation meeting requirements into the late 1990s.

REFERENCES

1. Crawford, D. J. and Cleveland, J. I. II, "The New Langley Research Center Advanced Real-Time Simulation (ARTS) System," AIAA Paper 86-2680, October 1986.

2. Crawford, D. J. and Cleveland, J. I. II, "The Langley Advanced Real-Time Simulation (ARTS) System," AIAA Journal of Aircraft, Vol 25, No. 2, February 1988, pp. 170-177.

3. Crawford, D. J., Cleveland, J. I. II, and Staib, R. O., "The Langley Advanced Real-Time Simulation (ARTS) System Status Report," AIAA Paper 88-4595-CP, September 1988.

4. Cleveland, J. I. II, Sudik, S. J., and Crawford, D. J., "High Performance Processors for Real-Time Flight Simulation," AIAA Paper 90-3140-CP, September 1990.

5. Cleary, R. T., "Enhanced CAMAC Serial Highway System," presented at the IEEE Nuclear Science Symposium, San Francisco, California, October 23-25, 1985.

6. ANSI/IEEE Standards 583, 595, and 675, Institute of Electrical and Electronic Engineers, 1976.

7. Bennington, D. R., "Real-Time Simulation Clock," LAR-13615, NASA Tech Briefs, June 1987.

8. Cleveland, J. I. II, Sudik, S. J., and Grove, Randall D., "High Performance Computing System for Flight Simulation at NASA Langley," AIAA Paper 91-2971-CP, August 1991.

9. Cleveland, J. I. II, "Application of Technology Developed for Flight Simulation at NASA Langley," presented at the Technology 2001 Conference, San Francisco, California, December 3-5, 1991.

10. Cleveland, J.I. II, "Use of Convex Supercomputers for Flight Simulation at NASA Langley," presented at the Convex Worldwide User Group Conference, Richardson, Texas, May 17-22, 1992.

BIOGRAPHY

Jeff I. Cleveland II is the Project Engineer of the LaRC Flight Simulation Computing System. He received a B.S. in Electrical Engineering in 1963 from Texas A&I University and an M.S. in Electrical Engineering and Computer Science in 1970 from The George Washington University. He has worked in the field of flight simulation and operating system software since 1963.

Steven J. Sudik is a Project Analyst with Unisys Corporation. He is the group leader for the Flight Simulation Computing System software development team. He received an A.B. in English and Philosophy from Marquette University in 1966 and an M.A. in Philosophy from Indiana University in 1971. Prior to coming to LaRC he held positions as a systems programmer and business applications programmer. He has worked at LaRC since 1975 in operating systems support.

Randall D. Grove is the Lead Simulation Engineer for the LaRC Flight Simulation Computing System. He received a B.S. in Applied Mathematics in 1963 from the California State Polytechnic College at San Luis Obispo. He has worked in the field of flight simulation and application program development since 1963 at LaRC.

TRADEMARKS

UNIX is a trademark of American Telephone and Telegraph.

CDC and CYBER are trademarks of Control Data Corporation.

Convex is a trademark of Convex Computer Corporation.

Flight Simulation Research Requirements

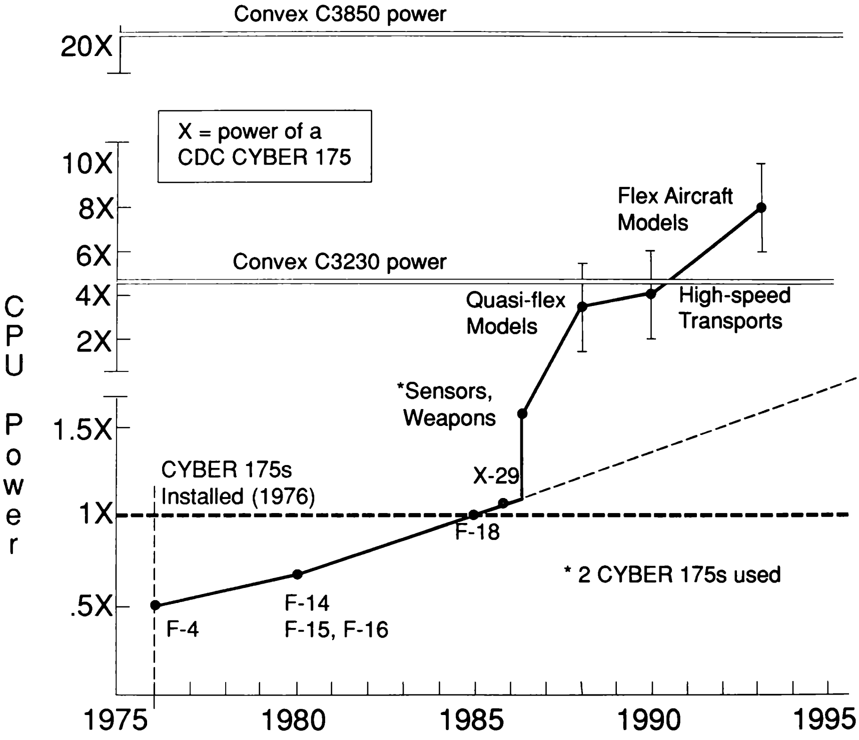
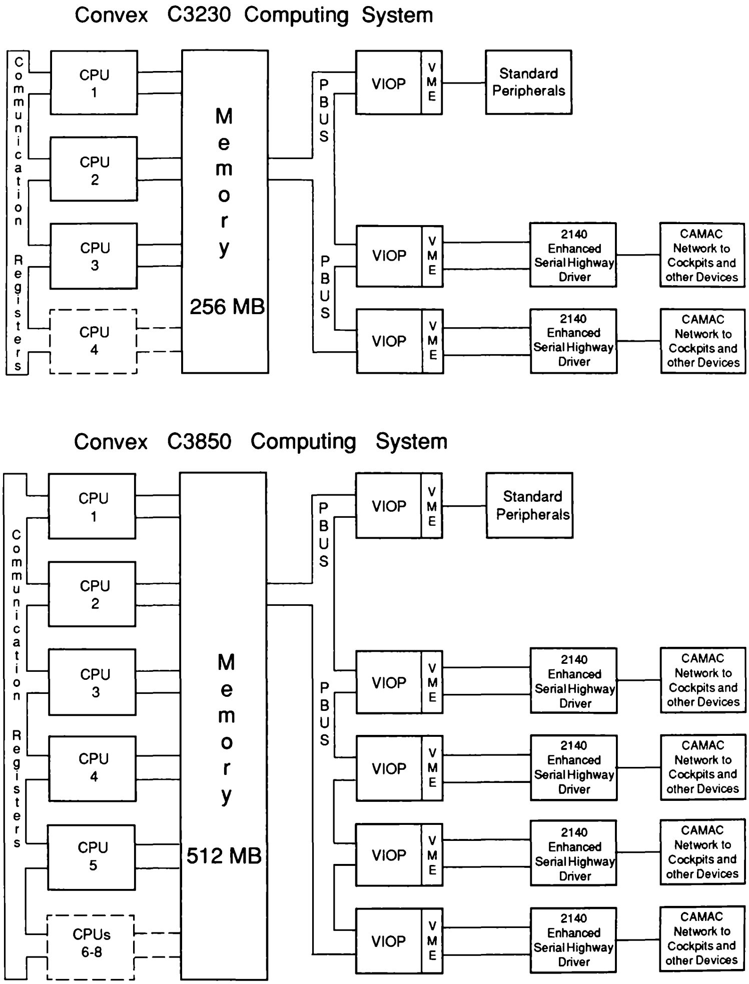


Figure 1.

Computing System Configuration


Convex C3230 Computing System

Convex C3850 Computing System

Figure 2.

ADDENDUM

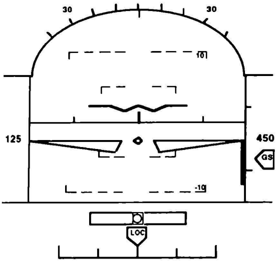


Fig. 3. Guidance Display

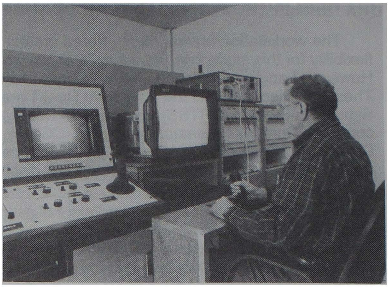


Figure 4. Pilot's Workstation

emulated imaging sensor display) of the runway and surrounding environment.

The guidance display (fig. 3) was presented on a CRT located on the turret assembly of the real-time control console (fig. 4). The guidance display presented a symbolic representation of the vertical situation to the pilot. It contained many of the standard symbologies found on electro-mechanical and "glass" flight director/attitude director type instrumentation. These included pitch and roll attitude, horizon, aircraft symbol, turn and bank indicator, glide slope and localizer scales. Several additional symbols or methods of presentation of standard symbols were included on the display. These included airspeed, altitude and altitude tape scale, vertical speed, and inertial flight path angle.

The visual landing image (fig. 2) was presented to the pilot on a 19-inch color CRT. Overlay masks were attached to the monitor to reduce the field-of-view for the different test cases. The resulting CT6 image was presented at unity magnification.

The pilot's "cockpit" (fig. 4) consisted of a small console upon which was mounted a three-axis hand controller, a slide potentiometer, and a discrete button switch. The three-axis hand controller, with spring loading and center detent, was used for pitch, roll, and yaw inputs. The slide potentiometer was used to represent the throttle control. The discrete button switch was used to change flap position. Each push of the button caused the flaps to move to the next flap setting. The test subjects stated that they had no difficulty in adjusting to this environment.

EXPERIMENT DESIGN

Five pilots of varying experience participated in this series of tests. They included a NASA test pilot, commercial airline pilot, a former military jet pilot, an instructor pilot, and a very experienced simulation design engineer. The pilots were given at least one training session that lasted approximately 2 hours. They were allowed to fly the simulated airplane in any manner that they desired for the first several minutes to get a feeling for the simulator's handling qualities. The remainder of the 2 hours was devoted to flying combinations of the specific conditions that the pilot would be flying during the test runs.

Forty-five test runs were used to evaluate the effects of the image FOV, magnification, aiding symbology, resolution, and update rate. The runs started at approximately 1000 feet above ground level about 5 miles from the runway. At the start of each data run, the aircraft was set up for straight and level flight and the heading was initialized randomly for each run to within 4 degrees of the runway heading. This procedure was used instead of adding turbulence to the simulation in order to force the pilot to exercise the lateral control axis of the airplane.

The pilot's task was to lower the flaps to the approach setting and slow to the approach speed of 125 knots, intercept the glideslope, maintain the aircraft on the glide slope and localizer, flare the airplane prior to landing on the runway, complete the touchdown, and remain on the runway centerline until the simulation run was ended by the console operator. Each pilot repeated each specific factor level three times. When possible, the order of variable presentation was randomly chosen within a repeat of each factor level.

The different fields-of-view were obtained by masking off the landing scene display with a paper mask (fig. 5). The full FOV of the monitor was set to 60 by 48 degrees. Five fields-of-view were evaluated, ranging from the smallest imaging sensor size anticipated to twice the size of conventional HUDs. The pilot was seated 13 inches from the monitor. The resulting fields-of-view were 5 by 5, 11 by 8, 14 by 11, 30 by 24, and 60 by 48 degrees. The fields-of-view were presented randomly within the group of five fields-of-view and repeated three times in different orders.

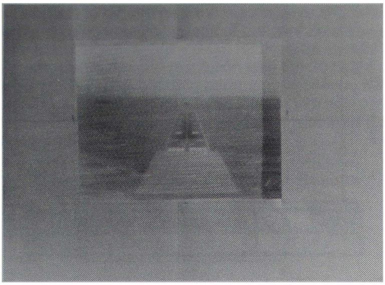


Figure 5. FOV Controlled by Masks

Magnification was controlled by programming the FOV in the CGI while maintaining a fixed distance from the monitor to the pilot. Aiding Symbology was overlaid graphically on the CGI image. Since the resolution of the monitor-CGI system was fixed, resolution changes were obtained by moving the monitor further from the pilot but maintaining an image of the same FOV with unity magnification to the pilot. Update rate was controlled by how often information was passed to the CGI system. Without new information the CGI system continues to redraw the previous scene.

RESULTS

The purpose of these tests was to determine which, if any, of five synthetic image display parameters (factors in the experiment) affect pilot landing performance. The results presented herein ensued from five sets of tests that evaluated each factor separately. The results will be discussed in the following order: FOV, image resolution, the use of aiding symbology with the raw sensor image, magnification of the sensor image, and image update rate.

Field-of-View

Five fields-of-view were tested (5 by 5, 11 by 8, 14 by 11, 30 by 24, and 60 by 48 degrees) with the display resolution set to 0.08 degrees per pixel, the update rate at 33 frames per second, and unity magnification. In order to understand the effects of the FOV on landing performance, it is necessary to evaluate what visual cues the runway provides the pilot during the landing flare. Figure 6 is a drawing of what the runway would look like to the pilot at the flare height of 50 feet. Superimposed on the figure are four of the fields-of-view tested. The edges of the runway at the aim point (this corresponds to the point at which the pilot will be looking - ref. 10) have been clipped by the smallest FOV (5 by 5) and is just about to be clipped by the next smallest FOV (11 by 8). Therefore, the pilot would not see the runway width expanding (visual flow field) at the aimpoint with these two fields-of-view. Figure 7 is a plot of the visual growth rate of the runway width at the aimpoint versus altitude. The curve is based upon an assumed constant airspeed and constant flight path. The altitudes at which the width of the aim point is clipped by four of the fields-of-view is indicated on the figure. The hyperbolic curve of growth rate is below visual detection at altitudes above 50 feet and, below 50 feet, exhibits a rapid increase in the growth rate. It can be hypothesized from these figures that the smaller fields-of-view will provide insufficient visual flow field cues for the pilot to initiate and complete the flare maneuver.

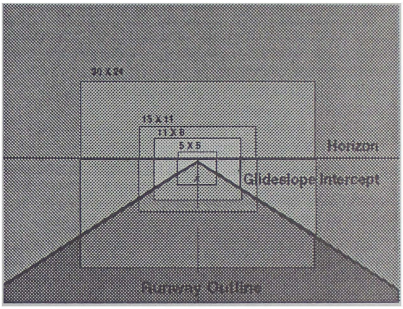


Figure 6. Runway at Flare Height

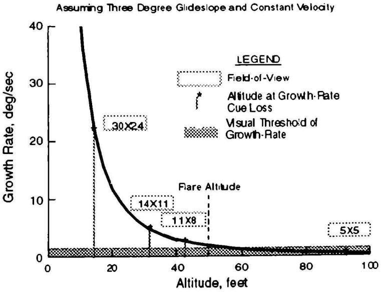


Figure 7. Runway Expansion Cues

Since the touchdown sink-rate performance parameter is known to be high in simulators, a flare index (FI) was developed (based upon the ratio of the reduction in sink-rate at 30 feet altitude to 75 percent of the desired reduction in sink-rate at

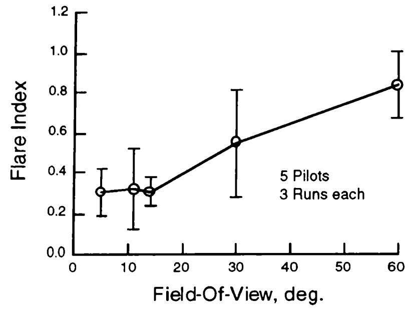


Figure 8. Flare Index (FI) versus FOV

touchdown).The FI measure was used to indicate how well the pilots were able to initiate the flare. A FI of unity approximates an exponential flare maneuver.

Figure 8 shows the resulting FI versus FOV. These data indicate that as the FOV increased the pilots were better able to initiate the flare. Statistically, there were two groups of fields-of-view. The smaller three fields-of-view were grouped together and the larger two fields-of-view were grouped together. These results confirmed the hypothesis that the smaller fields-of-view would degrade the pilot's ability to initiate the flare. Because of the good agreement of the data with the hypothesis, it is concluded that the pilot's use of the runway growth rate cue for flare initiation is valid.

A two-way repeated measures analysis of variance was performed upon the performance parameters to determine if the means were affected by the FOV factor. The following parameters were statistically significant ($p < 0.05$) for the FOV factor:

Table 1. **Analysis of Variance Results**

Performance Parameter	Sig. Level
Flare Index	0.017
Touchdown runway pos.	0.016
Touchdown airspeed	0.001
Heading Variation	0.005

A plot of the average touchdown parameters of airspeed and runway position for the five fields-of-view is presented in figure 9. As the FOV increased, the touchdown airspeed decreased and the touchdown position moved down the runway. This combination of these two factors is consistent with the increase in the FI during the flare maneuver. However, as anticipated, the sink rate at touchdown was not affected by the increase in FOV.

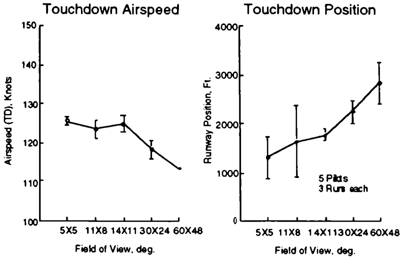


Figure 9. Touchdown Airspeed and Runway Position

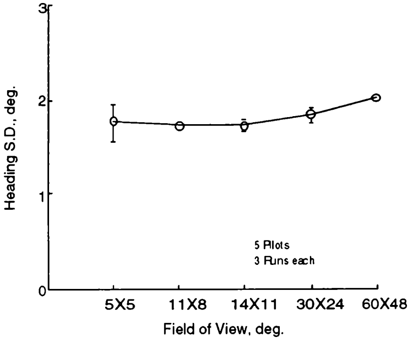


Figure 10. Approach Heading Standard Deviation

Paradoxically, the standard deviation of heading during the approach was lower with the smaller fields-of-view (fig. 10). In fact, pilot comments focused on the relative ease of keeping the runway centered with the smaller fields-of-view and harder with the larger fields-of-view. The effect was so noticeable that one pilot suggested having a variable FOV, narrow when far from the runway and wider when close to the runway for use in the flare. In terms of controlling heading, the 11 by 8 degree FOV condition was preferred by most of the pilots. However, as noted earlier the flare initiation was much better with the wider fields-of-view.

Image Resolution

Two pairs of image resolution levels were tested (0.08 versus 0.01 degrees per raster line at 11 by 8 degrees FOV and 0.08 versus 0.04 degrees per raster line at 30 by 24 degrees FOV). A two-way repeated measures analysis of variance was performed, for each FOV, upon the performance parameters to determine if the means were affected by the resolution factor. None of the parameters were statistically significant ($p < 0.05$) for the 0.08 vs 0.04 degrees per raster line resolution with the 30 by 24 degree FOV. The following parameter, however, was statistically significant ($p < 0.05$) for the 0.08 vs 0.01 degrees per raster line resolution factor with the 11 by 8 degree FOV:

Table 2. **Analysis of Variance Results**

Performance Parameter	Sig. Level
Touchdown Lateral Error	0.017

Error from the runway centerline is cut in half when the resolution is increased by a factor of eight (fig. 11). Although the FI was not statistically different between the 0.08 and 0.01 degrees per raster line resolution, the average FI did increase from 0.24 to 0.38 when the resolution was increased by a factor of eight (fig. 12).

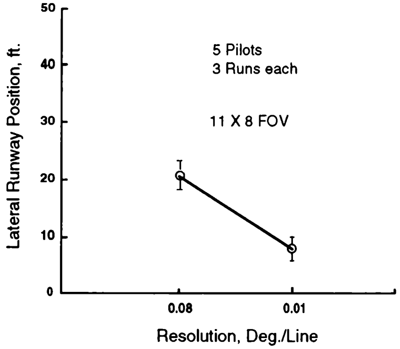


Figure 11. Lateral Error versus Resolution

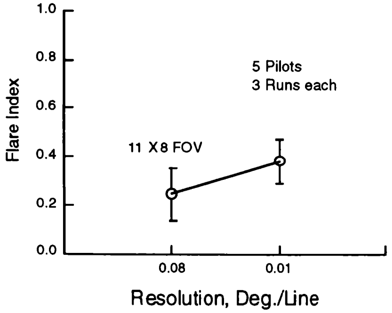


Figure 12. Flare Index (FI) versus Resolution

Subjectively, the pilots thought there was a difference in the 0.08 and 0.01 degrees per rasterline conditions. One pilot observed that the flare was smoother and another that it was easier to fly with the finer resolution. Another pilot observed that the finer detail of the image permitted more precise pitch control.

Aiding Symbology

Two conditions of aiding symbology were tested (with aiding symbology and without aiding symbology). The aiding symbology was essentially a simplified version of figure 3. No flare guidance cues were provided. A two-way repeated measures analysis of variance was performed upon the performance parameters to determine if the means were affected by the addition of aiding symbology factor. The following parameter was statistically significant ($p < .05$) for the aiding symbology factor comparing the 30 by 24 degree FOV with the 0.04 degrees per raster line resolution:

Table 3. **Analysis of Variance Results**

Performance Parameter	Sig. Level
Heading Variation	0.048

The heading variation parameter decreased from 1.9 to 1.6 degrees by adding aiding symbology (fig. 13). In the previous section the heading variation for small fields-of-view was about 1.8 degrees. Therefore, the aiding symbology is a better method of improving (lowering) heading deviation during a landing approach than using a small FOV.

Although it was not statistically significant, the Flare index increased from 0.43 without the aiding symbology to 0.55 (fig. 14) with the aiding symbology.

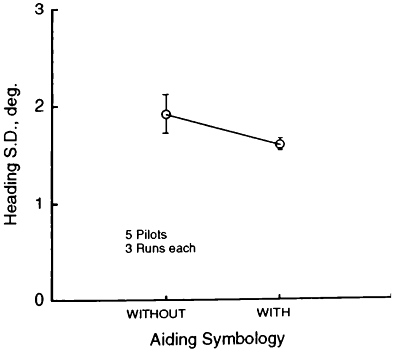


Figure 13. Approach Heading Variation

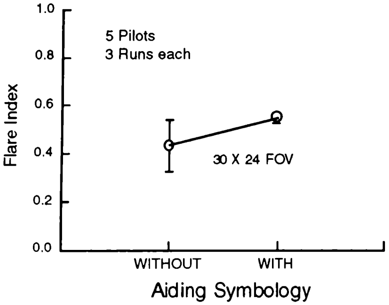


Figure 14. FI versus Aiding Symbology

Pilot comments concerning the aiding symbology noted the improvement in lateral performance. One pilot observed that there were more control inputs with the aiding symbology but that the inputs were smaller. However, during the flare the control inputs were fewer, resulting in a smoother flare. One pilot observed that the aiding symbology made the flying task so easy his "kid" could now fly the simulator. The aiding symbology was thought to be a very desirable addition to the landing display. The implementation of the symbology should be optimized, e.g., the position of the aircraft symbol should be moved so that it does not clutter the display. The aiding symbology causes a switch of emphasis in the pilot's mind, making the scene of secondary importance. Therefore, to perform the flare, the pilot had to switch concentration from the symbology to the runway.

It was anticipated that the aiding symbology would improve the touchdown vertical rate. Even though it did improve the flare performance, it did not change the touchdown vertical rate in this study. A flight test of a helmet mounted display having a closed-circuit image of the runway with aiding symbology produced identical touchdown vertical rates compared to visual landings (ref. 7). Since the visual image was of lower quality than normal vision it would appear that the use of aiding symbology was useful in maintaining performance with lower quality images.

Magnification of the Sensor Image

Five magnification factors were tested (0.75, 0.86, 1.00, 1.15, and 1.50). Since the monitor size remained constant, the resulting fields-of-view were 40 by 32, 35 by 28, 30 by 24, 26 by 20, and 20 by 16, respectively. A two-way repeated measures analysis of variance was performed upon the performance parameters to determine if the means were affected by the magnification factor. The following parameters were statistically significant ($p < 0.05$) for the magnification factor:

Table 4. **Analysis of Variance Results**

Performance Parameter	Sig. Level
Touchdown Sink-Rate	0.002
Touchdown Lateral Position	0.038
Heading Variation	0.002

Touchdown sink rate decreased as the magnification of the visual scene was increased (fig. 15). As noted in the previous section, the FOV changes did not influence touchdown sink rate, therefore, the differences in touchdown sink rate noted in these tests can be assumed to be the results of the increased magnification factor and not the concomitant reduction in FOV. The magnification factor can be thought of as a gain factor in the attitude axes. With the increase in pitch attitude gain the pilots were better able to control attitude and thus lower the touchdown sink rate.

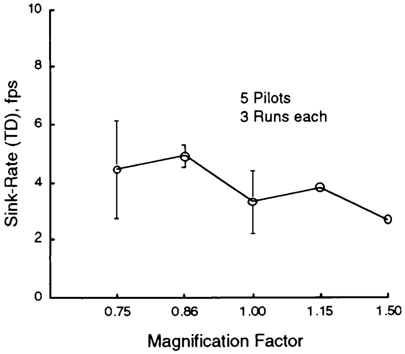


Figure 15. Touchdown Sink Rate versus Magnification

The magnification factor also affected the pilots ability to control the airplane laterally. The touchdown lateral position and the heading variation decreased with increasing magnification (fig. 16 and 17). However, as noted previously, the increase in magnification also lowered the FOV. As noted in the previous section, the FOV affected the heading variation parameter. Consequently, it is not immediately apparent whether the magnification alone or the reduced FOV alone or both factors caused the improvement in the lateral control. The amount of improvement of heading variation due to FOV (decreased from 30 degrees to 14 degrees) alone (see fig. 10) was approximately 0.1 degrees. From figure 17 the improvement in heading variation from 40 degrees FOV (.75 magnification) to 20 degrees FOV (1.50 magnification) was approximately 0.4 degrees. Therefore, it appears that the magnification factor improved the heading accuracy by almost a factor of four over the FOV effect.

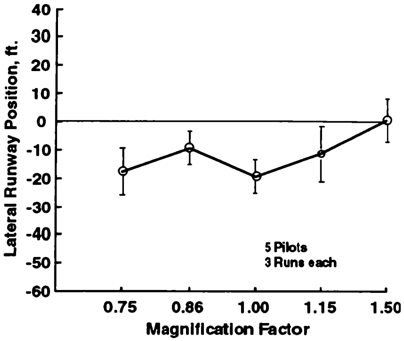


Figure 16. Touchdown Lateral Positions versus Magnification

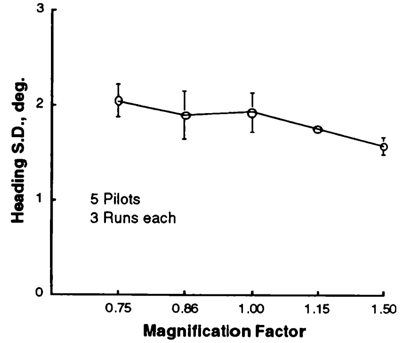


Figure 17. Approach Heading Variation versus Magnification

Pilots generally commented that the lower magnification factor resulted in higher workload and that they had to rely more on the auxiliary guidance display. One pilot felt that the magnification factor affected his apparent distance to the runway while a second pilot observed this as a change in the steepness of the approach angle and a third likened the lowest magnification factor to a no flap approach. One pilot suggested using a variable magnification during the approach by starting off with a magnification factor greater than one and reducing to unity magnification at the time of flare and touchdown. In general, the pilots did not like the magnification factors below one.

Previous research (ref. 11) had investigated the effect of magnification and had concluded that a magnification factor greater than one was needed for a contact analog display in order to compensate for the lack of cues in the display. These data would seem to support those conclusions and indicate that pilots are able to more accurately control sink rate and runway lateral error with magnification factor greater than unity. However, it is difficult to compare the present study with the previous study because of differences in the piloting tasks. The effects noted in this study by the pilots seem to be those expected due to the geometrical changes induced by the magnification and not due to short focal length of the eye compared to actual flight.

Image Update Rate

Four image update rates were tested (6, 11, 17, and 33 updates per second). A two-way repeated measures analysis of variance was performed upon the performance parameters to determine if the means were affected by the image update rate factor. The following parameters were statistically significant ($p < .05$) for the image update rate factor:

Table 5. **Analysis of Variance Results**

Performance Parameter	Sig. Level
Touchdown Airspeed	0.005
Touchdown Runway Position	0.005
Touchdown Sink Rate	0.036
Touchdown Lateral Position	0.018

All of the touchdown parameters were affected by the image update rate. The airspeed at touchdown was highest at the 6 frames per second update rate (fig. 18). The other update rates had the same touchdown airspeed. The touchdown runway position increased with image update rate (fig. 19). There was very little difference in the runway position for the 11, 17, and 33 frames per second update rate. The touchdown sink rate was highest for the 6 frames per second and lowest for the 17 and 33 frames per second update rate (fig. 20). Interestingly, the touchdown lateral runway position was on the centerline with the 6 frames per second update rate and left of center for the faster update rates (fig. 21). There seemed to be a tendency for the pilots to land left of center under most of the conditions. Whether this is because of their previous experience or not is uncertain.

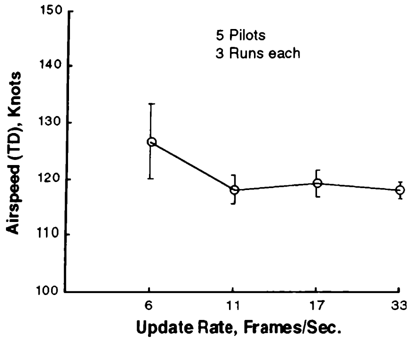


Figure 18. Touchdown Airspeed versus Update Rate

Subjectively, the pilots felt that the 6 frames per second update condition was jumpy. The 11 frames per second update condition seemed jumpy close to the ground. The two fastest conditions were not noticeably jumpy for the maneuvers used in these tests. One pilot commented that he could not tell if his lateral inputs were causing the proper response and that his judgment of his vertical inputs were much delayed with the 6 frames per second condition. Another pilot said that flaring was not an option with the 6 frames per second condition. The jumpiness in the peripheral vision with the 6 and 11 frames per second conditions were a little disconcerting. Another pilot felt that if he were flying the 6 frames per second update rate he would crash if he had any turbulence and that the 11 frames per second was marginally satisfactory. One pilot said that the 6 update per second condition had the same visual effect as flying a helicopter.

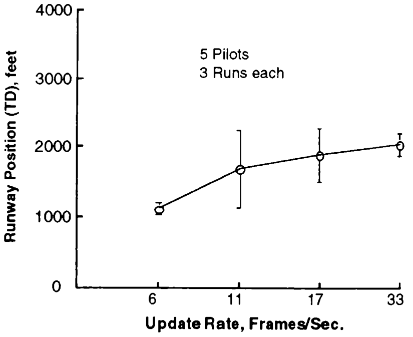


Figure 19. Touchdown Runway Position versus Update Rate

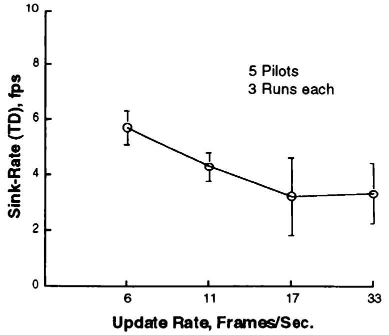


Figure 20. Touchdown Sink Rate versus Update Rate

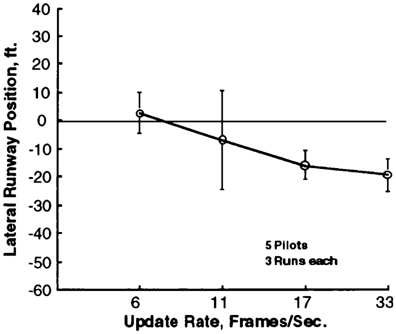


Figure 21 Touchdown Lateral Position versus Update Rate

DISCUSSION

The problems associated with flaring and landing an airplane using a synthetic vision display are the same problems the simulation community has had for decades trying to provide a synthetic visual scene which would allow the pilot to flare and land with performance comparable to the real world. Proper cues are needed in both cases to allow the pilot to perform satisfactory landings. Simulator scientists have tried providing several cues to the pilots in an attempt to obtain real world landing performance, but so far none has been totally successful. Cues that have been added include the following: collimated visual, color visual, two window visuals, texture within the visual scene, six-leg synergistic motion platform, g-seats, and helmet loaders. None of these have improved the flare and landing performance to the point of being comparable with the real world performance. The only method that seems to help is to provide the pilots with enough landing training so that they can learn to use whatever cues are available within the scene to perform satisfactory landings. One study indicated that over fifty landings are necessary to achieve a level of performance comparable to flight (ref. 12). The data in this report provides some clues as to what cues are missing as well as some insight into cues that should be provided in a synthetic display that pilots might use for landing airplanes.

FOV is important in providing the pilot the flare initiation cue. The streaming expansion of objects away from the aim point gets sufficiently rapid at the flare altitude that the pilot knows it is time to flare. In order for this cue to work properly, the pilot needs a certain amount of this streaming effect which is not limited by the FOV of the landing scene.

Touchdown sink rate was affected by magnification of the image. This improvement in touchdown sink rate was due to the easier/faster interpretation of motions resulting from pitch control inputs. One pilot suggested that one thing missing in this simulation was the simulation of the nose (or window edge) of the airplane. Use of the nose (or a grease pencil mark on the windscreen) has been used by pilots as a gage to use to judge pitch attitudes. A pitch reference line (a metal rod) was used in a lunar flying backpack study to provide the pilot with better pitch cues (ref. 13) since no vehicle structure was in front of the pilot. Perhaps the same phenomena was working in these simulations. The aim point was closer to the center of the screen than the bottom edge, therefore, slight movements were hard to differentiate with the wider fields-of-view. A similar result occurred in the heading standard deviation. With either the narrower FOV, increased magnification, or with aiding symbology the heading deviation decreased.

The cues most useful to control the flare may involve the perception of finer detail than is currently presented in simulation visual systems. This would suggest that even the level of texturing provided in current computer generated imagery is not detailed enough for the flare maneuver. The visual cue most often mentioned in reference 14 is the motion of one object relative to another object. This level of detail is present in computer graphic images for the large objects, but is generally not provided for the smaller objects.

One cue not exploited by the simulation community is the depth perception from stereopsis. This cue might provide the needed flare cues that would permit more accurate control of touchdown vertical speed. As noted previously from reference 5, the use of exaggerated stereopsis cues caused the pilots to flare high and to interpret their speed incorrectly. Stereopsis flare cues should be investigated.

CONCLUSIONS

The purpose of these tests was to determine which, if any, of five issues (factors in the experiment) affect pilot landing performance while using a video image of the runway and the surrounding environment. The five issues were field-of-view (FOV), magnification of the sensor image, image resolution, image update rate, and the use of aiding symbology with the sensor image.

The five fields-of-view used in these tests varied between 5 by 5 and 60 by 48 degrees. The FOV affected pilot performance in FI, touchdown airspeed and runway position, and heading variation during approach. The wider fields-of-view resulted in the initiation of flare maneuver to occur at the proper altitude and with more complete flaring. The role of the FOV in the landing flare is that wider fields-of-view provide enough of the streaming cues so that the pilot knows when to flare, but other cues are needed to control the flare maneuver precisely.

The magnification of the sensor image affected pilot performance in FI, touchdown sink rate, lateral position, and heading variation during the approach. Magnification of the image improved the pilot's ability to control the airplane attitude more precisely. The slightest motion of the attitude is detected and can be corrected. The pilots preferred unity magnification, possibly because of their desire to have it displayed on a head-up display and did not like for the image to be minified because of the change in cues that it provided.

Aiding symbology improved the FI and approach heading variation. The improvement in heading variation more than compensated for the degradation as the FOV was increased. Pilots preferred to use the aiding symbology and commented that it made the landing task much easier.

The image resolution was found to affect pilot performance in touchdown lateral position error. There were no differences found between 0.08 and 0.04 degrees per raster line but there were differences between 0.08 and 0.01 degrees per raster line.

The update rate of the image from 6 to 33 updates per second affected pilot performance in the following four touchdown parameters: airspeed, runway position, sink rate, and lateral position. The slower image update rate did not permit the pilots to perform a flare maneuver. The faster update rates produced much better flare and touchdown performance.

These five factors were found to affect landing performance and therefore, will have to be considered carefully in the design of any enhanced or synthetic vision system for aircraft. The resulting design will have to be a compromise of these factors which will provide for safe, acceptable performance of the landing maneuver by all pilots.

At least three visual cues present in the real world were not present in the workstation study. These cues could be important to pilots while performing flare and landing maneuvers. The three cues were: (1) very fine detail structure in the image, (2) relative motions in the visual scene of smaller objects, and (3) stereopsis cues.

REFERENCES

1. O. B. St.John: All Weather Landing. Shell Aviation News. No. 364, p. 2-11, 1968.

2. ANON: 'Blind' Plane Flies 15 Miles and Lands; Fog Peril Overcome. The New York Times, September 25, 1929.

3. John P. Reeder; and Joseph J. Kolnick: A Brief Study of Closed-Circuit Television for Aircraft Landing. NASA TN D-2185, February, 1964.

4. Bernard R. Kibort; and Fred J. Drinkwater, III: A Flight Study of Manual Blind Landing Performance Using Closed Circuit Television Displays. NASA TN D-2252, May, 1964.

5. Garrison P. Layton, Jr.; and William H. Dana: Flight Tests of a Wide-Angle, Indirect Optical Viewing System in a High-Performance Jet Aircraft. NASA TN D-3690, October, 1966.

6. G. M. Griffin, Ph.D.; and K. E. Matsunaga: External Vision Augmentation System. Eighth National Aero-Space Plane Technology Symposium, Naval Postgraduate School, Monterey, California, March 26-30, 1990.

7. Kenneth R. Yenni: Flight Tests of a Helmet-Mounted Display Synthetic Visibility System. AIAA-90-1279, AIAA/SFTE/DGLR/SETP Fifth Biannual Flight Test Conference, May, 1990.

8. Jack J. Hatfield; and Russell V. Parrish: Advanced Cockpit Technology for Future Civil Transport Aircraft. Eleventh Annual IEEE/AESS Dayton Chapter Symposium, November 28, 1990.

9. Staff of NASA Langley Research Center and Boeing Commercial Airplane Company: Terminal Configured Vehicle Program - Test Facilities Guide. NASA SP-435, 1980.

10. Amos A. Spady, Jr.: Airline Pilot Scanning Behavior during Approaches and Landing in a Boeing 737 Simulator. AGARD 25th Guidance and Control Panel Meeting/Symposium on "Guidance and Control Design Considerations for Low Altitude and Terminal Area Flight", October, 1977.

11. Stanley N. Roscoe: Flight by Periscope. University of Illinois Institute of Aviation Aeronautics Bulletin Number 9, 1951.

12. Anon: Approach and Landing Simulation. AGARD R-632, October, 1975.

13. Randall L. Harris, Sr.; Eric C. Stewart; Lee H. Person, Jr.; and Kenneth R. Yenni: Preliminary Research Flights of a One-Man Rocket Propelled Backpack in Simulated Lunar Gravity. NASA LWP-836, December 15, 1969.

14. Wolfgang Langewiesche: Stick and Rudder. McGraw-Hill Book Company, 1944 and renewed in 1972.

AUTHOR INDEX

(Number after name indicates session in which paper appears)